



# Reprojecting Visualizations for Advanced Interaction

DIPLOMARBEIT

zur Erlangung des akademischen Grades

**Diplom-Ingenieur**

im Rahmen des Studiums

**Medieninformatik und Visual Computing**

eingereicht von

**Sanjin Radoš**

Matrikelnummer 0226963

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Univ.Prof. Dipl.-Ing. Dr.techn. Eduard Gröller

Mitwirkung: Dipl.-Ing. Dr.techn. Krešimir Matković, VRVis Research Center in Vienna, Austria

Univ.Prof. Dipl.-Ing. Dr.techn. Helwig Hauser, University of Bergen

Wien, 8. August 2023

---

Sanjin Radoš

---

Eduard Gröller



Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar  
The approved original version of this thesis is available in print at TU Wien Bibliothek.



# Reprojecting Visualizations for Advanced Interaction

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

**Diplom-Ingenieur**

in

**Media Informatics and Visual Computing**

by

**Sanjin Radoš**

Registration Number 0226963

to the Faculty of Informatics

at the TU Wien

Advisor: Univ.Prof. Dipl.-Ing. Dr.techn. Eduard Gröller

Assistance: Dipl.-Ing. Dr.techn. Krešimir Matković, VRVis Research Center in Vienna, Austria  
Univ.Prof. Dipl.-Ing. Dr.techn. Helwig Hauser, University of Bergen

Vienna, August 8, 2023

---

Sanjin Radoš

---

Eduard Gröller



Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar  
The approved original version of this thesis is available in print at TU Wien Bibliothek.

# Erklärung zur Verfassung der Arbeit

Sanjin Radoš

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 8. August 2023

---

Sanjin Radoš



Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar  
The approved original version of this thesis is available in print at TU Wien Bibliothek.

# Danksagung

Ich möchte mich bei allen Menschen bedanken, die mich während der Diplomarbeit unterstützt haben. Zuerst möchte ich meinen Eltern **Ćamila** und **Ekrem Radoš** danken, die mich in allen Situationen liebevoll unterstützt haben. Umfangreiche Unterstützung kam auch vonseiten meiner Frau **Mirha**, die mich in schwierigen Situationen immer beruhigt hat. Ich möchte auch meiner Schwester **Edita** und meinen Kindern **Ahmed**, **Alima** und **Davud** danken, die mein Leben bereichern.

Diese Diplomarbeit wurde gemeinsam mit dem **VRVis Zentrum für Virtual Reality und Visualisierung Forschungs-GmbH** verfasst. Besonders möchte ich mich bei meinem damaligen Projektleiter und Freund **Krešimir Matković** bedanken, der mir die Chance gab, in seiner innovativen Forschungsgruppe zu arbeiten. Ein großes Dankeschön geht auch an meinen Betreuer **Meister Eduard Gröller**, der mich nicht nur bei der Diplomarbeit mit umfangreichen, konstruktiven Kommentaren geholfen hat, sondern auch wertvolle Ratschläge fürs Leben gegeben hat.

Ich danke Gott für all diese wunderbaren Menschen und auch für weitere, die ich hier nicht erwähnt habe, die es in meinem Leben gibt.



Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar  
The approved original version of this thesis is available in print at TU Wien Bibliothek.



# Kurzfassung

Die visuelle Analytik spielt eine immer wichtigere Rolle in der Datenexploration und -analyse. Bisher liefert ein erheblicher Teil der Methoden der visuellen Analytik vorwiegend qualitative Ergebnisse, beispielsweise basierend auf einer kontinuierlichen Farbskala oder einer detaillierten räumlichen Kodierung. Dies unterstützt und erleichtert die menschliche Beteiligung im Analyseprozess. Wichtige Anwendungen, wie etwa medizinische Diagnose und Entscheidungsfindung, profitieren jedoch von quantitativen Analyseergebnissen. Um die visuelle Analyse weiter zu stärken, schlägt diese Diplomarbeit mehrere Erweiterungen des etablierten Konzepts des Verknüpfens und Bürstens vor. Das Ziel ist die Erleichterung der quantitativen Interpretation von Ergebnissen aus dem Brushing und die Förderung der Reproduzierbarkeit dieser Ergebnisse. Wir tragen zur Reproduzierbarkeit bei, indem wir das Konzept eines strukturierten Brushing-Raums einführen, der Möglichkeiten für interaktive, quantitative und reproduzierbare visuelle Analysen bietet. Der strukturierte Brushing-Raum kann an die Nutzerpräferenzen angepasst werden durch spezifische Merkmale wie das hier eingeführte Prozentgitter und die Snap-to-Rasteroption für Brushing nutzt. Darüber hinaus werden zwei neue Brushing-Techniken vorgestellt: die Prozentbürste und die Mahalanobis-Bürste. Diese Techniken nutzen zugrunde liegende Daten, um statistisch sinnvolle Interaktionen mit Daten zu ermöglichen. Ein Beispiel ist die Auswahl eines festgelegten Prozentsatzes von Datenwerten, wie z.B. 10%. Überlagerte deskriptive Statistiken, die wir den verknüpften Ansichten hinzufügen, integrieren die quantitativen Ergebnisse, indem sie die Statistiken der gebürsteten Datenelemente aus anderen Dimensionen liefern. Das neue relative Differenzdiagramm unterstützt auch das Verständnis von Datenänderungen in den verknüpften Ansichten. Wir evaluieren die vorgeschlagenen Techniken für Analyseaufgaben, die reproduzierbare Ergebnisse und quantitative Messungen erfordern, im Kontext von zwei Fallstudien, und präsentieren Feedback von Fachexperten. Eine Studie basiert auf meteorologischen Daten und eine andere auf Daten über Länder der Welt.



Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar  
The approved original version of this thesis is available in print at TU Wien Bibliothek.

# Abstract

Visual analytics plays an increasingly important role in data exploration and analysis. Until now, a significant portion of visual analytics methods predominantly delivers qualitative results, for example, based on a continuous color map or a detailed spatial encoding. This facilitates and augments human involvement in the analysis process. Crucial applications, such as medical diagnosis and decision making, benefit from quantitative analysis results. To further enhance visual analytics, this thesis proposes several extensions to the well-established concepts of linking and brushing. The aim is to enable the quantitative interpretation of results from brushing and promote reproducibility. We address the reproducibility challenge by introducing the concept of a structured brushing space, offering opportunities for interactive, quantitative, and reproducible visual analyses. The structured brushing space can be tailored to user preferences by utilizing specific features like the introduced percentile grid and the snap-to-grid option for brushes. Additionally, two novel brushing techniques are introduced: the percentile brush and the Mahalanobis brush. These techniques use underlying data to allow statistically significant interactions. An example is to select a predetermined percentage of data items, like 10%. Overlay descriptive statistics that we add to the linked views incorporate the quantitative results by providing statistics of the brushed data items from other dimensions. The new relative difference plot also aids in understanding visualization changes in the linked views. We evaluate the proposed techniques for analysis tasks requiring reproducible results and quantitative readings in the context of two case studies. One is based on meteorological data and the other one on world countries data. We finally present feedback from domain experts.



Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar  
The approved original version of this thesis is available in print at TU Wien Bibliothek.

# Contents

<b>Kurzfassung</b>	<b>ix</b>
<b>Abstract</b>	<b>xi</b>
<b>Contents</b>	<b>xiii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation and Problem Statement . . . . .	4
1.2 Methodological Approach . . . . .	8
1.3 Contribution . . . . .	10
1.4 Outline of the Thesis . . . . .	11
<b>2 Scientific Context and Related Work</b>	<b>13</b>
2.1 The Value of Visual Analysis . . . . .	13
2.2 Interactive Visual Analysis . . . . .	30
2.3 Reproducibility within the Context of Visualization Research . . . . .	35
2.4 Enhancing Qualitative Analysis with Quantitative Information . . . . .	41
<b>3 Reproducible Brushing</b>	<b>45</b>
3.1 Structured Brushing Space . . . . .	45
3.2 Snap-to Options for Brushing . . . . .	49
3.3 Percentile Grid . . . . .	54
3.4 Grid Extensions for Parallel Coordinates . . . . .	56
3.5 Percentile Brushes . . . . .	59
3.6 Mahalanobis Brush . . . . .	70
3.7 Animated Brushing . . . . .	80
<b>4 Quantitative Linking</b>	<b>85</b>
4.1 Motivation . . . . .	85
4.2 Inclusion of (Descriptive) Statistics . . . . .	86
4.3 Relative Difference Plot . . . . .	104
<b>5 Demonstration</b>	<b>107</b>
5.1 Initial Use Case: Climate Data . . . . .	107
	<b>xiii</b>

5.2	Use Case: Exploring Statistics Data of Countries . . . . .	114
5.3	Related Publication . . . . .	118
<b>6</b>	<b>Discussion and Conclusion</b>	<b>121</b>
6.1	Research Objective IVA . . . . .	121
6.2	Human Factor Challenges in IVA . . . . .	123
6.3	Benefits of Constrained Brushing . . . . .	123
6.4	Strengthening IVA through Quantitative Information . . . . .	125
6.5	Discussion of the Evaluation . . . . .	126
6.6	Closing Statement . . . . .	129
6.7	Future Work . . . . .	130
	<b>Bibliography</b>	<b>131</b>

# Introduction

Effective means for data analysis are crucial to successfully exploit the wealth of information that is potentially contained in massive sets of complex (often heterogeneous) data. Because information is often hidden in data, applying only a single data analysis technique is often not enough to extract valuable insights. The scientific field known as visual analytics deals intensively with these problems. Visual analytics has realized that the path to success is to intertwine advances from different research areas—including statistics, machine learning, data mining, and visual analysis—to capitalize both on the perceptual and cognitive powers of users as well as on the efficiency of automated, large-scale computations [KMS<sup>+</sup>08]. Over time, visual analytics has evolved into a mature scientific field that has become an indispensable complement to automatic analysis techniques. Visual analytics supports both novice users and experienced analysts in performing data exploration and analysis efficiently and effectively. It can facilitate the analysis of data in both simple and complex analytical processes by providing the means that not only help the user to monitor and interact with the graphically presented data, but also to influence the different phases of the process. The great freedom and power to extract insights from the data would not be possible without the help of various visual interaction techniques adapted to work with the human visual system. Many of these techniques have been developed in the sub-field of visual analytics known as interactive visual analysis (IVA), which we will learn and contribute to in this thesis.

IVA provides an efficient and effective framework in which analysts can take full advantage of the power of human perception and cognition as well as their expertise in researching and analyzing the data in question. Integral parts of this framework are the means for interaction and data visualization. For the visual communication of different aspects (dimensions) of the data, IVA uses many visualization techniques, such as a parallel coordinate plot, scatterplot, curve display, and bar chart, to name a few. Data visualization techniques used in IVA are traditionally made interactive, i.e., they provide a mechanism by which the user can modify graphically presented data (that is why we

often hear the term interactive visualizations). The most popular interaction technique used in IVA is known as brushing (read selecting). Brushing was introduced many years ago [BC87], and the original intent of brushing was—and still is—to select and visually emphasize or highlight brushed data items in different views. Many different brushing techniques have been developed over the years, but research into new techniques is still highly needed because different data types, visualizations, and use cases create demands and needs for new or improved brushing techniques. To enable complex data analysis, IVA uses a well-proven approach known as coordinated multiple views (CMV), in which multiple visualizations are used to visualize different data dimensions jointly, and users can correlate those views [Rob07]. The basic idea of CMV is to enable interaction with data in all coordinated views through brushing. When the user brushes the data items of interest, the linking mechanism is immediately activated. It will update all the linked views and make the selected data subset visually emphasized (for example, through different color coding) so that the user can notice and observe related changes in other dimensions of the same data set. Brushing combined with linking within the setup of the coordinated and multiple views is better known as linking&brushing. With the inclusion of linking&brushing, additional possibilities for understanding data and searching for hidden information are activated [Mun14].

An example of data analysis using linking&brushing in CMV is shown in Figure 1.1, where six dimensions of the meteorology data set [NOA14] are visualized. At the beginning of the visual analysis process, we usually choose which dimensions of the data we want to explore or analyze. In the example shown, two scatterplots jointly display the elevation and maximum temperature (top-left) and the latitude and longitude values (top-right). Histograms at the bottom display the recorded maximum amount of precipitation (bottom-left) and the minimum amount of precipitation (bottom-right). By observing visualizations, the user creates a mental image of the graphically presented data, which eventually leads to the “I see (something)!” effect. For example, there is one small cluster of dots in the lower-left corner of the left scatterplot with very low elevation and relatively low maximum temperature values. After gaining the first insight, the user typically wants to explore “this something” further. For example, he might be interested in finding the geographic locations of the measuring stations that make up the aforementioned cluster. Hence, the user initiates data exploration by creating a brush to select the data subset of interest, for example, in the left scatterplot, as shown in Figure 1.1.

The CMV system responds to the user interaction by activating the linking mechanism, which immediately and consistently highlights the associated subset of the data items in all linked views. Because the user knows the relation of the brushed data in the brushed view he can now explore and analyze the respective data subspace in the linked views. The scatterplot on the right in Figure 1.1 reveals that the measuring stations in question are located near the coast, but are relatively distant from each other. It also shows that the brushed subset of the data no longer forms a compact cluster, as in the brushed view, but is divided into several smaller clusters. Furthermore, as revealed by histograms





Figure 1.1: ComVis [MFGH08] a coordinated multiple views system is used to analyze meteorology data. ComVis links all its views, when a subset of the brushed data changes in the brushed view. In the shown example, a brush is created in the top-left view and the graphical representation of the data is immediately updated in all other views.

at the bottom, there is an observable difference in the precipitation measured at the selected stations. The user may at any time suspend or continue further analysis of the data. By repeating or creating new brushing operations, the user enters into an interactive&iterative dialogue with the analytical system. This is known as the IVA loop and helps him focus his analysis and perform a deep(er) information drill-down.

In the example shown, the next step in the analysis could be to create a new brush in the scatterplot on the right to examine how geographical location is responsible for the amount of precipitation, or the user could keep brushing the left scatterplot to examine other data subsets of interest. Linking&brushing is an essential feature in visual analytics systems, where it serves as a primary mechanism for interactive visual exploration and analysis of data.

## 1.1 Motivation and Problem Statement

The basic idea of linking&brushing in coordinated and multiple views (CMV) is that the user moves the brush in one view and watches what is happening in other views. Because he knows what is changing in the brushed view, he observes the consequences of it in the linked view(s). In this regard, the user performing the interaction plays a central role. However, designers of interactive visualizations often do not consider the user's behavior and the burden of his task. The user is most heavily burdened by cognitive loads that he or she has to invest. Unfortunately, these are often unnecessarily high, and when designing interactive visualizations, the possibilities for reducing the interactive load on the user are, in most cases, not sufficiently taken into account. In the context of linking&brushing in CMV, there are two aspects to consider that have a significant impact on the user's cognitive load. On the one hand, in the brushed view, i.e., where the brush is located, the user must take care of controlling the anchoring, extent, and positioning of the brush. When the user moves the brush, he does so in a targeted manner in order to observe the changes in the linked views, as exemplified in Figure 1.2. However, the relations he observes are often challenging to comprehend at first glance, which is why he has to repeat the brushing process several times in a row, for example, by following the previous brush path as much as possible. Being in control over the brushing operations is essential here, as it allows the user to select data subsets of interest more efficiently, move the brush precisely, and, at the same time, makes it easier for him to describe the brush and interpret the data selected by the brush. On the other hand, once the user is aware of what he has done in the brushed window, he can dedicate himself to the other side, i.e., to the linked views, where he must interpret the changes resulting from his brushing operations. Thus, to take full advantage of the power of the linking&brushing technique, the user must concentrate well on controlling the brush and understanding the changes in the visualizations. Since visual analysis commonly yields qualitative results, this further increases the user's cognitive load, especially if decisions have to be made on the fly, based on the user's interpretations of interactive visualizations.

This thesis helps users of linking&brushing in CMV, by supporting them both, on the brushing side where the interaction with the data is happening and on the linking side where the resulting changes are observed. In concrete, we provide new means that enable precise control of the brushing operation for making brushes that can be easily described and reproduced and we show additional information about the brushed data for an easier and more quantitative understanding of the visualization results. Our goals match the demands addressed by Kandogan et al. [KBHP14] in an interview with business analysts. Below we briefly explain the nature of problems we solve with our work.

By its design, visual analysis predominantly yields qualitative results—based, for example, on a continuous color map or detailed spatial encoding. The data to be explored and analyzed, as well as analysis results, are primarily visualized, i.e., encoded, with a combination of visual cues that are positioned, scaled, and colored according to the data values. The main reason for mapping data to geometry and color is that humans are

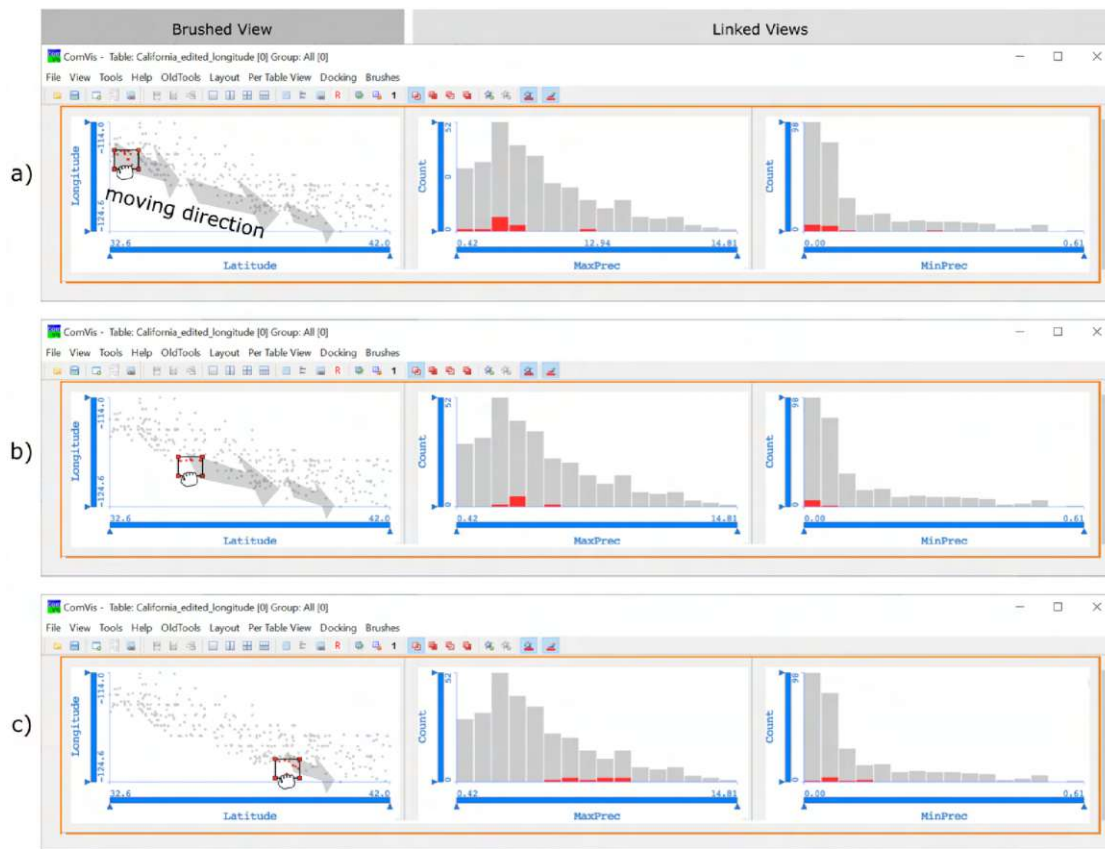


Figure 1.2: Illustration of a simple visual analysis using linking & brushing in CMV: The task is to analyze the precipitation along the California coast. Three images (a, b, c) are displayed for three different brush positions in the scatterplot, i.e., in the brushed view. The direction of movement of the brush in the scatterplot is indicated by arrows so that the reader can better follow the user's intention (in reality, it is a continuous movement of the brush). The user moves the brush in the brushed view and, knowing that he is moving it from north to south along the coast, he observes changes in the amount of precipitation that is displayed in two linked histograms.

highly visual. In carefully designed visualizations, humans can quickly recognize patterns and make visual comparisons of graphically presented data [Wil96]. An example how visual cues can quickly convey important messages about the visualized data is given in Figure 1.1. Histograms rely on the height of the bins to encode the number of data items belonging to a particular bin. The entire range in which the temperature fluctuated throughout the year is divided into 16 sub-ranges, i.e., bins, and the higher the bin, the more data there is in that bin. Scatterplots make use of labels and positions of dots in 2-D space to represent values for two different numeric variables. The scatterplot on the right shows the geographic location of weather stations in a country and the

scatterplot on the left communicates the relationship between the elevation and the largest measured temperature at different measuring stations in California. The qualitative character of visual analytics is essential because it enables the combination of human knowledge, intuition, and perception with the power of modern computers to support swift visual analysis and the rapid discovery of knowledge from an increasing amount of data. Although this “visual method of analysis” helps the user gain valuable knowledge quickly, it provides only approximate readings from the resulting qualitative visualizations. However, quantitative results are highly valued or necessary in many areas of science and real life, for example, when deciding on a medical diagnosis. Therefore, additional quantitative information would undoubtedly be helpful to many users performing interactive visual analysis to better and more accurately interpret the brushes and data being brushed in the linked views. Despite the clear need to improve the traditional visual analysis with additional quantitative information about the brushed data, most modern visual analysis tools do not have this capability and rely solely on providing qualitative insight. Even when there is a possibility of presenting quantitative information, in most cases, it will only be through a mere table that the user can activate on demand. Such a table usually has a fixed number of columns and provides an overview of precomputed data attribute values. Often it misses summary statistics and temporal evolution of a statistical value calculated from the brushed data. Also, the use of the data table to provide dynamically changing values can even distort the fluidity of the analysis as users may be forced to frequently change their focus between the data table and views in which the data is brushed or relations are observed. It is, therefore, an additional requirement to display the desired quantitative value in a better-positioned place, i.e., as close as possible to the data it describes. An example uses quantitative overlays within the observed visualization itself so that the user has the necessary information at hand without additional cognitive effort.

Interactive visual data analysis also suffers from a major weakness in terms of not being sufficiently reproducible. Supporting reproducibility of the results is an important condition for the widespread acceptance of visual analytics as a standard method for data exploration and data analysis. So far, the problem of reproducibility has been discussed more in other areas of science than in the visualization community [FF20]. The need for reproducibility of results is expressed everywhere—it is often said that science advances through corroboration, which means that it is of great importance that researchers can reproduce or verify other people’s results. However, depending on the situation, different users may have different requirements for the reproducibility of the results. One can require resources that allow the reproducibility of the entire analytical process, while at the same time, the other may want to repeat only a specific part of the analysis, or the most recent step of the analysis such as the last-performed brushing operation. As explained briefly above, brushing is a standard technique in many IVA contexts for highlighting, selecting, or deleting a subset of data items of interest, and solving an analytical task using IVA will almost certainly involve at least one brushing operation that the user performs to complete various actions on the visualized data. The basic brushing process involves the following operations: creating a new brush, resizing it, and

moving the brush around the visualization to select new data items. At first glance, it seems that the user can easily reproduce these basic brushing operations. But it can be very difficult or almost impossible due to, for example, the high resolution of modern screens and the fact that the user mainly uses the freely controlled mouse to carry out the brushing process. Even minor variations in brush placement can change the subset of brush data, as shown in Figure 1.3. This can lead to a significant change of related results in the linked views. There is need for efficient brushing mechanisms that provide users of visual analysis systems an advanced control over brushing operations, including means for repeating brushing operations or reproducing the brushing results. A typical example where reproducibility is required is repeating the brushing process after the results have been saved in a report or as shared laboratory notes. A note could consist of the following information: “Dear colleague, after selecting 10% of the lowest values of dimension X in the left scatterplot (please see the attached image), I discovered in the right scatterplot in the third quartile of dimension Y an unexpected horizontally-elongated cluster with the center value at (0.705, 1.980). Please repeat my analysis tomorrow and let me know if the cluster changes after the data file is automatically updated with new values overnight.” To the best of our knowledge, the “simple” analysis described above can not be easily reproduced with little time and effort, for example, with a few clicks of the mouse and using commonly available tools for visual analysis.

Additional problem we aim to solve is that when performing a very useful analysis based on rank, IVA users cannot meet their needs due to the lack of tools that support such an analysis. Note that there is a difference between brushing, for example, a 10% interval on an axis (which is a value-based analysis), and 10% of all data items shown (which is a rank-based analysis). In both cases, the user wants 10%, but the difference is whether to select all data items that correspond to a particular range of values or select a certain number of data items. IVA tools support value-based analysis per default. IVA uses human perception as a high-speed filter, trained to work well with a numerical scale on an axis. Users are able to do value-based analysis fairly accurately in standard visualizations such as a scatterplot without the need to apply or create additional filters. However, customized implementations are required to support a rank-based analysis. Including both possibilities by default will potentially help users to increase the extraction of insights from the analyzed data.

Users also need help to brush specific structures in a scatterplot that are elongated or have an angular orientation. With traditional brushing the brush has a precise interval defined by the user, i.e., an exact specification of the x-axis and y-axis intervals for the brush extent is given. If the user is brushing, for example, with a rectangular brush in a scatterplot, he knows precisely—mathematically—that he has brushed from  $x_1$  to  $x_2$  and from  $y_1$  to  $y_2$ . However, if the brush would take into account the underlying data distribution, it would help the user to save time and more quickly create new brushes that select elongated structures.

To summarize, extensions to visual analysis, which enable reproducible and quantitative results, may become key to further strengthen deployment of interactive visualizations in

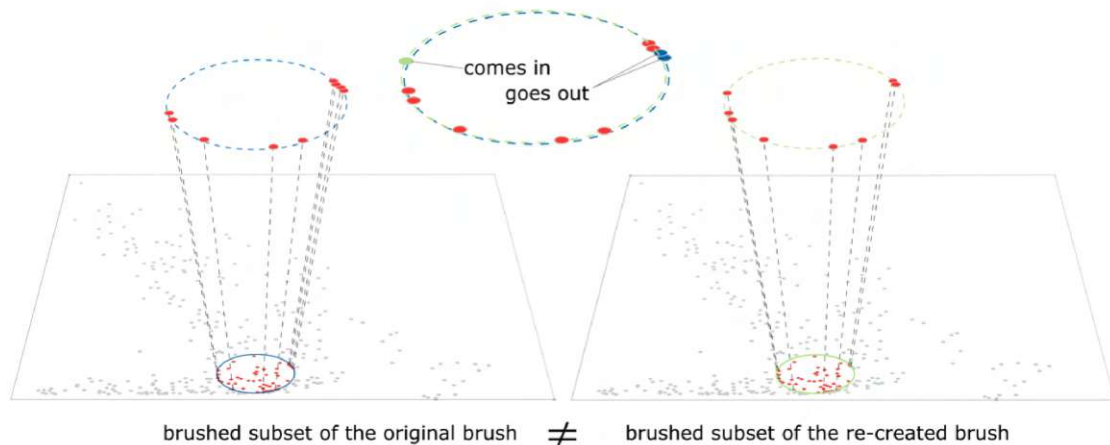


Figure 1.3: An attempt to reproduce a circular brush in a scatter plot. The direct manipulation takes place directly on the data display with a mouse as a pointing device. **Left:** The brush is placed in the desired location to select a subset of data items. Items near the border or at the border of the brush are shown enlarged. **Right:** The user has created a new brush with the goal to select the same subset of data items. **Top-Middle:** The outlines of the two brushes are shown. Blue color is used for the original brush and green color is used for the reproduced brush. Due to the slight difference in the placement of the brushes, the selected subset in the left scatterplot does not match the selected subset in the right scatterplot. The reproduced brush selects an additional item (shown in green), but two items (shown in blue) are no longer selected.

data analytics applications. Our idea is to introduce semi-structured brushing space and extensions for the linked views that can help IVA users to more easily select data subsets of interest, recreate their selections, and quantify them.

## 1.2 Methodological Approach

This thesis contributes to two important research questions in the context of interactive visual analysis, namely how to enable the reproducibility of brushing results, and how to deal with the lack of quantitative information in linked views. In general, the reproducibility and interpretation of visual analysis results are two issues that are treated separately in the visual analysis community. However, both issues are critical to linking&brushing in CMV, and therefore, we address them together in this work.

First, we examine the literature on general reproducibility problems in visual analytics to gain insight into problems associated with reproducibility related to brushing techniques. This is an insufficiently researched topic and decided to contribute here. The problem of difficult reproducibility of results from brushing in interactive visualizations is mainly related to the impossibility of precisely controlling the brush by hand. One feasible



solution to this problem is to introduce the facility that enables precise control of the anchoring, extent, and movement of the brush. This idea led us to the concept of a semi-structured brushing space, which proved general enough to be applied to a number of existing brushing techniques and visualization types.

We also examined the relevant literature on the interpretability of interactive visual analysis results and found this to be a hot topic nowadays. Most of the research only considers the qualitative aspect of visual analysis, despite the fact that many analysts demand quantitative information on top of the already useful qualitative information. This is reported by Kandogan et al. [KBHP14] based on 34 in-depth interviews in the context of business intelligence. We looked further and found that some older works from the nineties had already conducted some research to include quantitative values in the visual analysis [HBC<sup>+</sup>91]. Then we decided to revise and incorporate some of these long-ago suggested techniques into current state-of-the-art interactive visualizations.

We could not cover every visualization type—the number of existing interactive visualizations is enormous, and new ones are frequently emerging. Therefore, we decided to consider the two very commonly used ones: a 2-D scatterplot and a parallel coordinates plot. Both plots have a long history of utilizing brushing techniques [BC87, RLA<sup>+</sup>19] and are usually used to convey new ideas. In both views, we experiment with quantitative overlays and propose suggestions to improve quantitative readings.

Furthermore, current brushing techniques implemented in a scatterplot, such as a rectangular brush, do not provide an easy way to for selecting data items belonging to elongated and angled structures. To support this, we introduce the Mahalanobis brush for a scatterplot that is based on the Mahalanobis distance [Mah36]. The Mahalanobis brush considers the underlying data and changes its shape automatically to capture coherent structures.

Taking into account the underlying data distribution opens up new brushing options. We also experiment with brushing opportunities that enable a rank-based analysis in addition to a standard value-based analysis. To support both analysis variants, we expand the standard brushing possibilities by introducing new percentile brushes, which select a certain number of items specified by the user. We implement the rectangular percentile brush and the circular percentile brush in a scatterplot. In parallel coordinate plots, we added a one-dimensional percentile brush.

Moreover, in addition to adding quantitative values, which support decision-making, we developed a new qualitative visualization technique that makes it easier and quicker for the user to understand the relative changes in linked views resulting from a selection change in a brushed view. We call the new visualization a *relative difference plot*. As the name suggests, relative changes are emphasized on top of absolute deviations, e.g., a change of the center and the spread of a selected subset.

All research and development of the techniques presented in this thesis has been carried out in ComVis [MFGH08], an interactive visual analysis application that was developed at the VRVis research center in Vienna [VRV]. The key feature for choosing ComVis is the

ability to quickly develop new interaction and visualization techniques and incorporate them into a system of coordinated and multiple views. The complete development was performed in accordance with the basic principles of interactive visual analysis, with particular attention being paid to the sustainability of the fluid interaction between the user and a visualization system.

To ensure the proposed techniques meet the interactive visual analysis requirements for working with multidimensional data, we use one of the often cited data sets during research and development, i.e., the California meteorological data [NOA14]. This data set contains numerical and categorical data and includes several weather measurements, such as precipitation and temperature values recorded at 300 weather stations in California, including their geographic location and elevation. Our decision to use meteorological data is based on the fact that readers from different backgrounds can easily interpret it and devote their full attention to understanding the techniques described here. In terms of used data types, we decide to focus on numerical data only because visual analytics deals with such data more often than with categorical data. Users typically want to see different statistical measurements such as mean, median, and midrange, and these can be quickly calculated from numerical values. We follow the agile software development principle since we intend to integrate a comprehensive set of techniques into the ComVis tool. After developing and integrating the core concepts, we conducted a demonstration to gather user feedback, which we utilize to determine the need for improvements in the design. The final demonstration was scheduled upon the completion of the work, after all implementations had been finished. Here we use a different dataset in order to cover the aspect of generalization as well.

### 1.3 Contribution

As we strive to analyze larger and larger amounts of complex and multidimensional, often heterogeneous, data, interactive visual exploration and analysis becomes increasingly important and remains an ongoing research topic. This thesis introduces a set of relatively simple but effective and efficient extensions to IVA, where an overview is shown in Figure 1.4. They contribute to solving the two practical problems explained in Section 1.1. The contributions can be summarized as follows: we

- show how to improve the reproducibility of results from brushing by introducing the concept of a semi-structured brushing space. It is based on controlling the anchoring, the extent, and the movement of the brush. The proposed concept does not violate the existing control freedom of the user over the brushing operation. Instead, it increases the possibilities for controlling the brush by only influencing certain aspects of the brushing operation that the user has intentionally constrained. The intention is to help himself to control/interpret the brushing actions better and reproduce the results of these actions later more easily,



## Extensions for Reproducible & Quantitative Visual Analytics

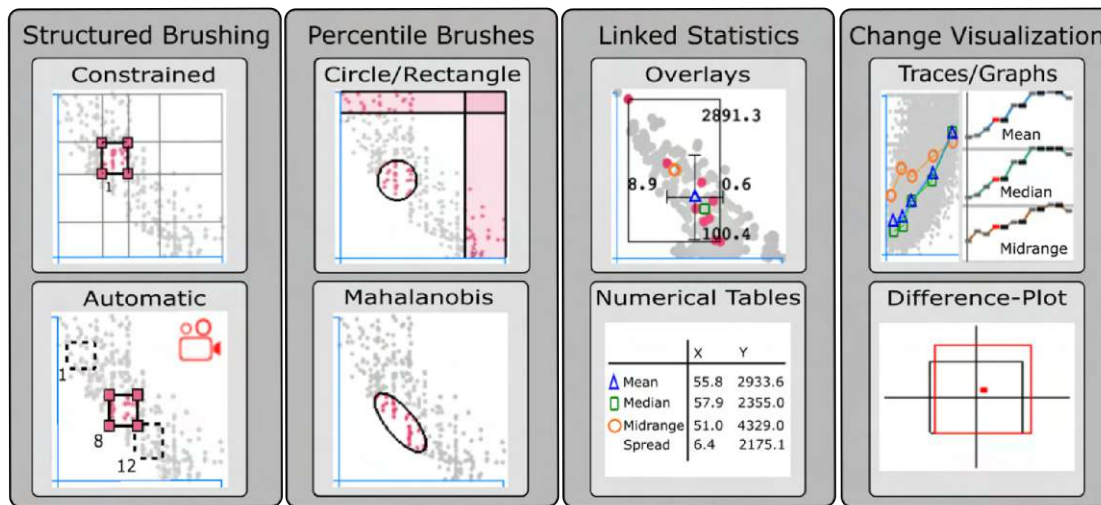


Figure 1.4: An overview of new extensions for IVA that were presented in this thesis.

- introduce two novel brushing techniques: the percentile brush (based on statistics) and the Mahalanobis brush (considering the underlying data distribution),
- provide new ways for rank-based analysis. This analytical approach is supported by newly introduced percentile brushing techniques, as well as by structuring the brushing space using a percentile grid,
- propose animated brushing as another approach to aid reproducibility and interpretation of the linked views,
- present different overlays as a way to integrate quantitative results into visual analytics on top of the more common qualitative results to support decision making,
- introduce the relative difference plot, a new way to support understanding data changes in linked views. This plot emphasizes relative changes on top of absolute deviations.

## 1.4 Outline of the Thesis

Chapter 2 provides scientific context and related work. The reader should start with Section 2.1 if he is unfamiliar with visual data analysis. We give a historical overview of information visualization and explain the value of visual analysis for exploration and analysis of multidimensional data. Section 2.2 introduces the basic concepts of interactive visual analysis, which are necessary to comprehend the following chapters of the thesis. In Section 2.3 we discuss

the state-of-the-art techniques concerning reproducibility and in Section 2.4 the quantitative interpretability of results from interactive visual analysis.

- Chapter 3 proposes new solutions to the problem of brushing reproducibility. The model of the brushing space is presented in Section 3.1. An example of a possible solution for (*partially*) *constrained* brushing using snap-to-grid option is given in Section 3.2. The percentile grid for a scatterplot is introduced in Section 3.3, while Section 3.4 discusses snap-to-grid options for a parallel coordinates plot. This chapter also discusses rank-based analysis and data-aware brushing. As a concrete solution, the percentile brush is introduced in Section 3.5, and the Mahalanobis brush in Section 3.6, respectively. Moreover, the animated brush is proposed in Section 3.7 as another way to enable support the reproducibility of results from brushing. This is an example of (*semi*-)*automatic* brushing.
- Chapter 4 presents further extensions, including the integration of descriptive statistics, which enables a quantitative reading of linked views with focus+context visualization. Section 4.1 explains the importance of providing statistical values about the brushed data, and Section 4.2 shows how descriptive statistics and traces from brushing can be used to help analyze data at different trace position as a way to improve the data discovery process and reduce the time to insight. Furthermore, the relative difference plot presented in Section 4.2 is as a novel way of describing the history of linked data statistic.
- Chapter 5 demonstrates the usefulness of the new techniques for interactive analysis of multidimensional data. A preliminary demonstration and feedback from visual analysis experts are briefly described at the beginning of the chapter in Section 5.1. Then in Section 5.2, details are given for the final demonstration of the successful use of our new technology in the context of a study of ensemble data from climatology. Section 5.3 briefly presents results from a research paper that implements the techniques presented in this thesis.
- Chapter 6 finally concludes the thesis by summarizing what we have learned and discusses the advantages and disadvantages of the current approach and presents selected ideas for future work.

Parts of this thesis are based on the publication: S. Radoš, R. Splechna, K. Matković, M. Đuras, E. Gröller, H. Hauser: Towards Quantitative Visual Analytics with Structured Brushing and Linked Statistics, Eurographics Conference on Visualization (EuroVis) 2016 [RSM<sup>+</sup>16].

# Scientific Context and Related Work

Today, our knowledge-based society strives for a complete datafication of our world. The ability to produce, collect and store data is increasing faster than the ability to analyze it. Various reasons lead to the need for a comprehensive analysis of data, for example, to support the development of a cure for new diseases, as is currently the case with COVID-19 [PNH<sup>+</sup>20]. The explosive growth in the amount of raw data exacerbates challenges in various scientific disciplines, including visual analytics. As early as 2006, Keim et al. [KMSZ06] recognized and discussed these problems. We still witness remarkable efforts in researching tools and methods for data analysis that facilitate the transformation of “big” data into valuable information. In this respect, interactive visual analysis (IVA) is very useful. It is a sub-field of visual analytics specialized in analyzing data with many different data dimensions and many data items. IVA advocates the visual representation of data combined with means that allow people to interact with the data. In the remainder of this chapter, we provide an overview of the scientific context and related work in this field. In Section 2.1 we give an introduction to data visualization and visual analysis that can help readers from other fields to more easily understand the value of IVA. In Section 2.2 we give a summary of the interaction techniques used in IVA. Section 2.3 discusses the research work on the reproducibility of analysis results. We close this chapter with Section 2.4 which outlines the research toward more quantitative visual analysis.

## 2.1 The Value of Visual Analysis

The term visualization is nowadays used in different contexts, and therefore it has been defined many times with many different aims. One of the oldest definitions describes visualization as the process of creating a mental image. Another one that applies well in

the context of visual analysis says that visualization is the generation of images using a computer for the purpose of understanding data. The most accepted definition, which also fits the subject of this thesis, comes from Card, Mackinlay, and Schneiderman; they describe visualization as the use of computer-supported, interactive, visual representations of (abstract) data to amplify cognition [CMS99].

The advantage of visual analysis is that it transforms raw data into visible forms by graphically presenting data, helping humans gain insights into the data. This is possible because humans interpret visual information more efficiently than tabular data (data conveyed in table form) and because we have enormous visual bandwidth and unprecedented capabilities to quickly process what we see. Our visual sense constitutes about 90% of our perception, and with our eyes, we are literally sampling the world around us, including graphically presented data. Within the blink of an eye, a plethora of aspects of our world are processed, and we make sense of what we see. For example, we immediately can recognize how far away an object is that we want to take with our hand or what is the size of objects in the spatial 3-D layout of our surroundings. Since we are so much dependent on them, signals from the eyes travel ten times faster than signals coming from the touch [Nø98]. Further, we learn new skills through consistent repetition and practice, and thus we can become especially good at discovering meaningful patterns in visualized data. For example, trained visual analysis experts are preattentively able, to recognize similarities, deviations, trends, clusters and other patterns in visualizations. We concentrate in this work on interactive visual analysis. It is a still underrepresented field, which has recognized that visualization is such a powerful amplifier of human abilities and that by providing the proper mechanisms for interacting and manipulating graphically displayed data, it can offer users invaluable new ways to explore and analyze data.

### 2.1.1 Historical Examples

Since the dawn of humankind, humans have recognized and utilized the power of visualization to communicate messages. Long before the advent of computers and graphical displays, the visualization of information (on a medium such as paper) was used as an eminently shareable form of information that was otherwise complex to describe. As early as in the 15th century Leonardo da Vinci drew sketches of natural phenomena that helped him conduct research more efficiently and communicate his discoveries to others. His sketch of a free water jet is shown in Figure 2.3a and is one of the world's first flow visualization representations. However, it was only a few hundred years ago that people had enough data at their disposal, for example, from economic trading, observing weather changes, and war reports, and enough acquired statistical understanding to do thorough data exploration and analysis.

The advancement in visual thinking and data visualization can be traced back to people like Johann H. Lambert (1728–1777), William Playfair (1759–1823), Florence Nightingale (1820 – 1910), and Charles Joseph Minard (1781 – 1870), to name a few. Rapid advances in data visualization have accompanied the need to investigate sensible comparisons such as the combination of different variables and their relationships. For example, during

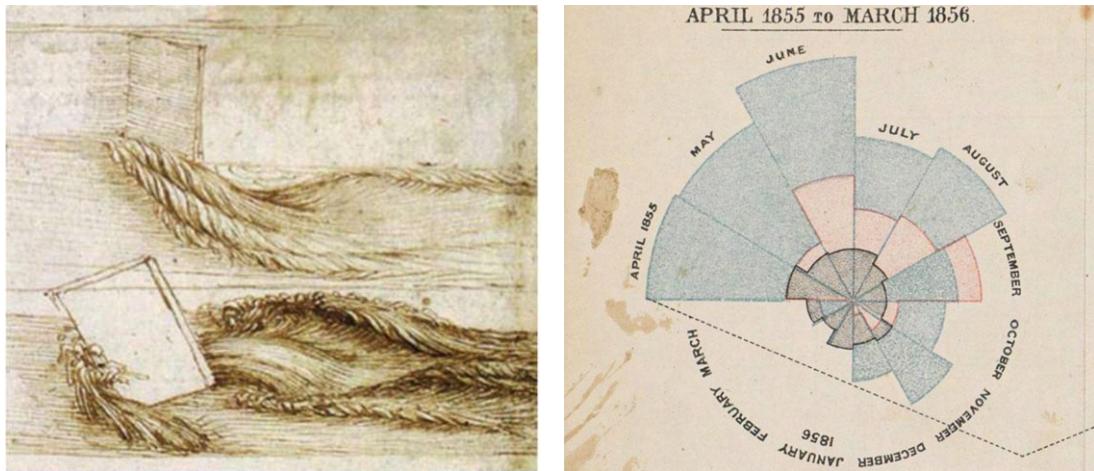
the Crimean War (1853 – 1856), Florence Nightingale, a nurse, and a statistics and data visualization pioneer, realized that there could be a correlation between high soldier mortality in hospitals and poor sanitation. Based on the collected data, she created a set of remarkable and original diagrams (Figure 2.1b shows one of them) that revealed a terrible picture. Many more soldiers died from diseases, occurring mainly due to poor conditions in hospitals, than from the direct consequences of injuries in combat with the enemy. With this visual evidence in hand, she convinced the Queen and the military leadership of the need to develop better human care for the sick and wounded.

Another example comes from Charles Joseph Minard, who was an expert in showing numerical data in the context of the spatiotemporal domain. His hand-made cartographic maps convey carefully weighed and relevant information in an exciting and informative way. One of his most famous works, shown in Figure 2.1c, communicates essential facts related to the French army’s invasion of Russia in the war of 1812. Although this is hand-made cartography, it includes a graphical representation of as many as six variables. Minard paid the utmost attention to the visualization design to avoid clutter and highlight important information. For this, he used different visual encodings, such as the thickness of the band to show the size of the army at specific geographic positions during the advance and retreat, different band colors to indicate a direction, with black color used for the path of retreat, and a series of thin vertical gray lines to accentuate the low winter temperatures during the retreat from Moscow. The lines intersect the path of the army at specific positions and temperatures are displayed in the timeline graph on the bottom. In addition, he provides comparisons, for example, we can see that Napoleon started the invasion with 420 000 French soldiers. Only the remnants of the army came back, as shown by the comparison given on the far left of the map, where the tan and black lines intersect.

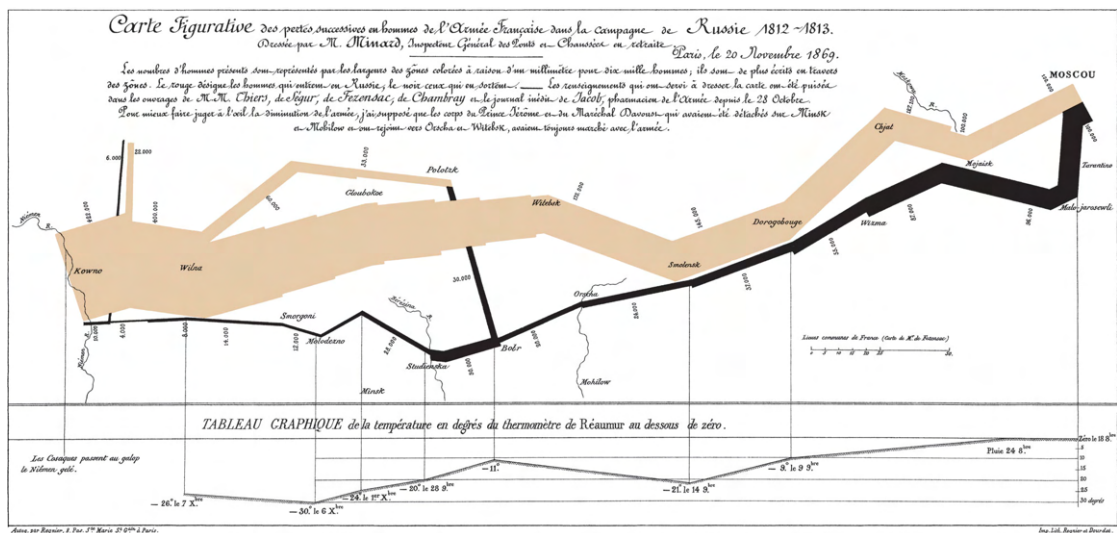
As many other beautiful examples demonstrate, a great data visualization design leads to beautifully designed stories that can explain data in seconds. However, because of the conflicting viewpoints about the importance of details in visualization designs, it took a long path before the scientific community reached a consensus on the purpose of visualization. Not so long ago, graphs created by statisticians were overloaded with quantitative information, making such representations hardly understandable outside the community. At the same time, graphs made for the masses, such as those in newspapers, often refine or distort the facts using “pretty” visualizations with many distracting elements that are not really relevant for information communication. Edward Tufte, a veteran of visual thinking and analytical design, refers to these superfluous elements as chartjunk [Tuf83]. There is also a comment by Ben Schneiderman on this, who says, “the purpose of visualization should be insight, not images”. For a detailed overview on the history of data visualization, see Tufte’s extensive work [Tuf] and the most recent book on this subject by Friendly and Wainer [FW21].



## 2. SCIENTIFIC CONTEXT AND RELATED WORK



(a) The freehand drawing of the movement of the water behind a solid obstacle [dV]. (b) Diagram of the causes of mortality in the British army during the Crimean War [Nig].



(c) The losses suffered by Napoleon's army in the Russian campaign of 1812 [Min69].

Figure 2.1: Hand-drawn visualization examples. (a): During his research into natural phenomena, Leonardo da Vinci (1452 – 1519) often drew sketches to help understand. (b): Florence Nightingale (1820 – 1910) enhanced a polar-area diagram to show that more soldiers died from disease (shown in blue) than from wounds (shown in red) during the Crimean War. Black encodes all other causes of death. Time constitutes an intrinsic function here, i.e., the diagram shows counts of deaths by month. (c): The cartography created by Charles Joseph Minard (1781 – 1870) gives us more than simply visual information; it communicates the narrative of the army's casualties, from the gathering point near Kaunas in modern-day Lithuania, towards Moscow and then back to home.

### 2.1.2 From Data to Insight

We are immersed in a sea of (numerical) data, and humans depend on techniques that can extend their reach and transform data into valuable insights to drive their businesses forward. That is often a difficult challenge due to various circumstances, such as increased complexity and variability of the data. For example, today in medicine, it is common to have a high number of data sets concerning the same phenomena, with data coming from different sources, such as a combination of anatomical, computed tomography, and functional data. According to Reinsel et al. [RGR17], the amount of worldwide available stored digital data will reach a total of 163 zettabytes by the year 2025, from 16.1 zettabytes in 2016. Gaining insights from the data starts with figuring out what users want from their data. For example, if a fisherman has a full fishing net, he probably wants to know how many kilograms of fish he caught in total, what is its volume so that transport can be properly organized, how many fish he caught that are much heavier than the average and that he could sell directly to a restaurant and so on. For a fisherman's simple analytical task, it is sufficient to calculate descriptive statistics related to the catch using automatic analysis methods, and he can immediately use the quantitative results to make data-driven decisions. Still, different users will have different needs, and different data will require different techniques.

Back in 1977, John Tukey emphasized that visualization can “force us to notice what we never expected to see” [Tuk77]. At the time Tukey said this, the prevailing opinion among statisticians was that numerical calculations are accurate and sufficient, but graphs are rough, meaning that they provide a visual, often simplified, and sometimes approximate representation of data trends and patterns. Although the fact is that the numbers are more precise or exact, data can contain hidden information that is not, or cannot be, conveyed with numbers extracted from the data. We may need the help of data visualizations in addition to doing statistical analysis, as explained below.

Francis Anscombe, who was one of the statisticians who saw the benefits of visualization, argued that data visualization could be crucial for understanding the relationship between variables. For demonstration, Anscombe constructed a well-known data set known as Anscombe's quartet [Ans73]. It comprises of four different synthetic data sets, each with eleven  $(x,y)$  items, as shown in Table 2.1. All four data sets share nearly identical simple descriptive statistics, as shown in Listing 2.1. There are undoubtedly additional statistics that have been intentionally excluded in order to avoid indicating differences between the data sets of Anscombe's Quartet. One example is kurtosis, which measures the deviation of a distribution's shape from that of a normal distribution. Looking only at the calculated numbers in Listing 2.1, it could be concluded that if they were visualized with the  $x$  and  $y$  axes, the shape of points for all four data sets would look very similar. Nevertheless, this is not the case, and Figure 2.2 reveals that the basic patterns of the four data sets are very different in shape when visualized. There is another interesting detail related to the graphical representation of the Anscombe's Quartet. The graphs of the data, if shown alone (see Figure 2.2), will not reveal that the four data sets have the same basic statistical profile. Anscombe's example clearly demonstrates that depending

on the analytical task and the data itself, a mixture of computational and visualization methods may be needed to understand the data.

Table 2.1: Anscombe's Quartet.

I		II		III		IV	
X	Y	X	Y	X	Y	X	Y
10.00	8.04	10.00	9.14	10.00	7.46	8.00	6.58
8.00	6.95	8.00	8.14	8.00	6.77	8.00	5.76
13.00	7.58	13.00	8.74	13.00	12.74	8.00	7.71
9.00	8.81	9.00	8.77	9.00	7.11	8.00	8.84
11.00	8.33	11.00	9.26	11.00	7.81	8.00	8.47
14.00	9.96	14.00	8.10	14.00	8.84	8.00	7.04
6.00	7.24	6.00	6.13	6.00	6.08	8.00	5.25
4.00	4.26	4.00	3.10	4.00	5.39	19.00	12.50
12.00	10.84	12.00	9.13	12.00	8.15	8.00	5.56
7.00	4.82	7.00	7.26	7.00	6.42	8.00	7.91
5.00	5.68	5.00	4.74	5.00	5.73	8.00	6.89

Listing 2.1: Statistical characteristics of Anscombe's Quartet:

mean of the x values = 9.0

mean of the y values = 7.5

equation of the least-squared regression line is:  $y = 3 + 0.5x$

sums of squared errors (about the mean) = 110.0

regression sums of squared errors = 27.5

residual sums of squared errors = 13.75

correlation coefficient = 0.82

### 2.1.3 Human Factor

By visualizing data, we paint a picture of the data conveying a clear idea of what the data means mostly with the intention to tell a story, inspire action, or to better understand the data. Data visualization, particularly the field of visual analysis, has become increasingly popular in recent decades. One reason is that it greatly supports the extraction of insights from data. Another reason for its high acceptance among different users is that visual analysis is accessible to everyone. Most of us naturally possess the ability to decode graphical data representation and think about it without requiring advanced knowledge of, for example, statistical or automatic analysis techniques. In other words, thanks to human visual perception which is jointly connected with thinking, visualized data is relatively easy to grasp. Our eyes are not just an input mechanism, but often solve problems through an appropriate visual representation. Often, to see a pattern is to



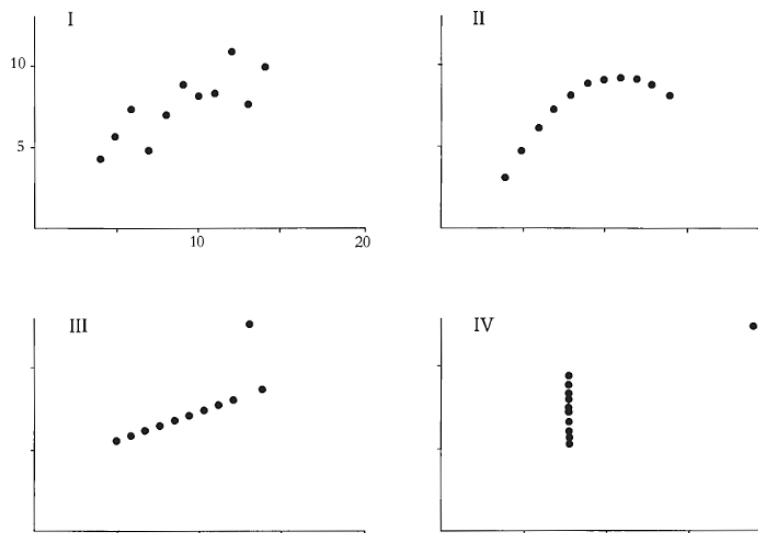


Figure 2.2: Visualization of Anscombe's Quartet [Ans73]. We perceive the data through its visual representation, where we see the patterns.

understand the solution to the problem [War22]. Understanding of how human visual perception works is of major importance for making better visualizations. The interplay between human perception and visualization is discussed in great detail in the book *Information Visualization: Perception for Design* [War22] by Colin Ware.

In the context of visual data analysis, data is usually presented to the user using appropriate graphical elements on a computer screen to support encoding of visualizations in our brain [CMS99]. That includes, for example, recognizing patterns, shapes, colors, and other visual elements to extract meaning or insights from the visual representation. For example, in Anscombe's demonstration (see Figure 2.2), data items are visualized using points as graphical elements in a scatterplot, where a dot represents a single data item  $(x, y)$ . Anscombe chose this type of plot because it is convenient to show relationships between two numeric variables, unexpected data gaps, and outliers. The visual system helps to immediately recognize that, for example, there is a simple linear relationship between data points shown in the top-left scatterplot, and there is no correlation in the bottom-right scatterplot.

The person who best recognized the power of visualization as a means to “see” what data can convey beyond the results of automated analysis techniques was John Tukey. Since 1977, he has passionately advocated a new approach to statistics called *exploratory data analysis* [Tuk77]. He suggested a whole new set of ideas on the use of visualization in data exploration, for example, *PRIM-9* [FFT75] is the first program to use interactive, dynamic graphics for viewing and dissecting multivariate data, encouraging developments in a wide range of fields and areas such as mathematics, statistics, computing, and informatics. All of this has led to the modern visual analysis of data that we know today.

We mean the same throughout the remainder of this work when we speak of *visual data analysis* and *visual analysis*. Nowadays, statistical plots, infographics, charts, and other computer-supported data visualization methods are used in visual analysis, for example, to help users explore data, identify their structures and find valuable inherent insights. Regardless of the goal, visualization certainly has to be an engaging and memorable experience so that it conveys a good amount of understanding of the data to the human observer. As Eduard Tufte notes, eyes can make a remarkable number of distinctions within a small area – it just has to be provoked to do so [Tuf83]. Tufte also found that visual representations of data could be more precise and revealing than conventional statistical computations [Tuf83]. Ben Shneiderman said that visualization even provides answers to questions the user did not know he had [Shn]. However, as noted by Blei and Smyth [BS17], it may be vital to include domain knowledge to fully understand, analyze, and interpret actual phenomena with data. Moreover, Robert Kosara [Kos] comments that visualization changes how data are understood and increases interest in data, thus encouraging more and better data to be generated. Leonardo da Vinci was among the pioneers in using visualization as a tool to understand natural phenomena. If, instead of creating rudimentary sketches on paper, he had the opportunity to simulate the water flow and observe the visualized results on the screen, as shown in Figure 2.3, his discovery would have progressed much faster.

### 2.1.4 Choosing the right Visualization

Because the visual representation of data is essential in many scientific fields, many visualization experts are concerned with creating guidelines that lead to clarity, precision, and efficiency in visualizations. For example, Tamara Munzner [Mun14] formalized the requirements for the analysis and the design of effective visualization methods in terms of principles and design choices. In her other influential work on design methodology [Mun10], Munzner presented a four-level nested model for visualization design and validation that addresses both visualization researchers and visualization designers. At its highest level, her model starts with the domain situation, then goes through the abstraction level and the idiom level, all the way to the algorithm. The output from a level above is input to the level below, so bringing attention to the design challenge that an upstream error inevitably cascades to all downstream levels. In the book *The Visual Display of Quantitative Information* [Tuf83], Tufte discusses the theory behind visualization methods, points out bad practices, and gives design guidelines. One such guideline is Tufte’s famous “data-ink ratio” which states that the largest share of ink on a graphic should present data information. Accordingly, non-data-ink is the ink that does not convey data-information but is used, for example, for scales, labels, background, and edges. In his book, Tufte clarifies how visualizations shape and many times distort—for example, through intentionally or unintentionally inserted lie factors—our understanding of the depicted information. As Mayr et al. [MHSW19] note, in order to increase the user’s perception of trust in the visualization, one should increase the trustworthiness of the visualization by including all relevant information. Matthew O. Ward also warned that misinterpretation of the data due to an inadvertent data distortion is often a problem

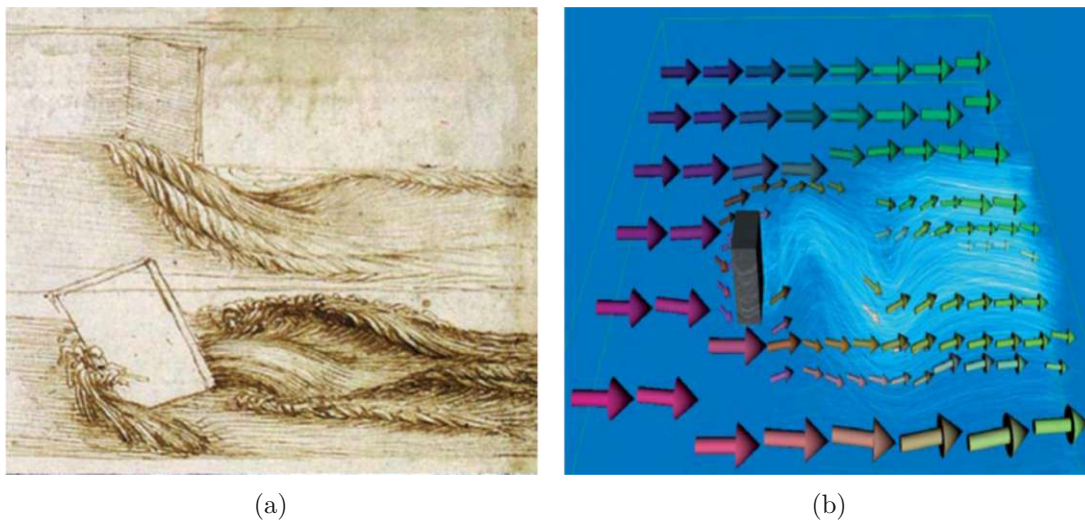


Figure 2.3: The movement of water past a solid obstacle. **(a)**: a hand-drawn sketch by Leonardo da Vinci. **(b)**: a computer-aided importance-driven visualization by Bürger et al. [BKKW08]. The explicative visual description of the flow at different levels of detail was created in two steps. First, streamlines were drawn to indicate areas of turbulent flow where local coherence is low. Second, glyphs were used to show regions of more stable flow; both their size and orientation vary as the coherence value increases.

in designing effective visualizations [WGK15]. For example, 3-D should be used only for representations of true 3-D data (e.g., volumetric data) and not as a third dimension of depth in bar or pie charts where it does not add additional information but instead creates distortions. The reader can find a collection of all kinds of bad design choices in the book *How Charts Lie* by Alberto Cairo [Cai19]. The enormously varied nature of data and users make it unavoidable to sometimes violate the guidelines, of course, if doing so serves the purpose of the intended information representation.

Using different techniques, visualization designers can ensure the inclusion of important data attributes, such as space and time, within a visualization. This aids users in comprehending complex data and drawing conclusions. The wide variety of available visualization techniques for data reasoning can discourage a new user from using data visualization. For example, the user works with univariate data, i.e., with just one attribute (or data dimension). In this case, he is limited to specific types of charts in the sense that he can only use charts that can display a single data dimension. Figure 1.1 shows an example of two bar charts, each visualizing a different dimension of the same data set. However, if the user is working with multidimensional data, he also has a multitude of charts available to choose from, but the question he asks himself is what type of visualization is best to use. For example, he may know that a 2-D scatterplot is the most commonly used visualization type in statistics to visualize two data attributes in a single plot. However, the question remains whether the scatterplot is the right plot to convey the message about his data at hand. The visualization community has

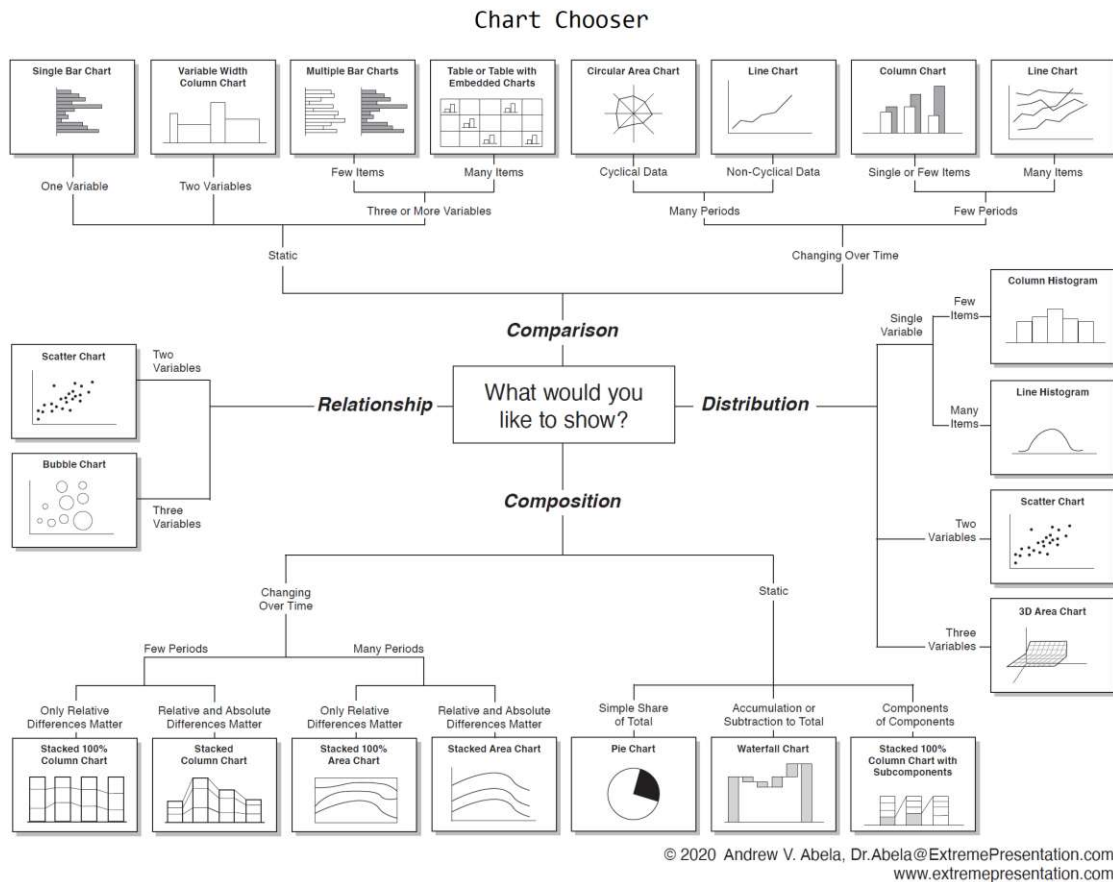


Figure 2.4: Chart Chooser [Abe13]. An example of a diagram that helps decide which charts are better suited for given data and problem.

recognized this problem and has addressed it through many projects and books focused on explaining charts and helping the user define which chart to use for a specific task and data. Cleveland’s book *Visualizing Data* [Cle93], and Wilkinson’s book *The Grammar of Graphics (Statistics and Computing)* [Wil05] are good references for getting familiar with all kinds of visualization methods and the fundamentals of how data and graphics connect. In her book *Visualization Analysis and Design* [Mun14], Tamara Munzner consolidated and refined existing concepts and terminology in visualization and associated common data types and user requirements with well-known visualization methods. Also, there are several great sources on the web that can help the user find the proper data visualization method for his data, such as *Data Visualization Catalogue* [Cat] and *From Data to Viz* [tV]. Another practical example is the “Chart Chooser” displayed in Figure 2.4 which first asks the user to decide on what he wants to show, whether it is comparisons, distributions, compositions, or relationships in the data, and then it shows the appropriate chart based on data attributes. However, picking from a set of pre-designed charts does

not provide flexibility. Pre-built charts, such as simple scatterplots and bar charts, are great when they meet users' needs. Many users, including researchers, analysts, and journalists, require additional customization of their charts to communicate, not just data values but also the context and intention. Users good at programming can relatively easily—using some charting library, like plotly [plo], bokeh [bok], or D3 [BOH11]—design carefully constructed and highly customized visualizations. For those who are not too willing to learn how to write code, there are some excellent environments with a focus on interactive authoring and design flexibility that support all users in quickly designing custom visualizations. Examples include Lyra [SH14], ECharts [LMS<sup>+</sup>18], and Charticulator [RLB19] that even works as an app for Power BI [pb]. For a more complete list, see the survey by Mei et al. [MMWC18].

Moreover, due to the falling prices of immersive technologies, we are witnessing an increased use of 3-D visualizations, especially those displayed on head-mounted displays. These new technologies allow users to immerse themselves into their data and interact with them using humans' visual and haptic perceptions. A recent overview of existing literature regarding visualization in virtual reality, reports extensive research efforts in developing immersive visualizations for various problem domains. However, there have been insufficient efforts to build a theoretical background and explore new methods of interaction beyond those provided by controllers [KS22]. Although 3-D visualizations and accompanying technologies bring challenges, they also open a promising new field of opportunities for different domains. For example, see the work by Traxler et al. [THC19] that enables data-driven navigation in 3-D for immersive tunnel monitoring.

To explain the proposed ideas in our work, we decided to augment parallel coordinates plots and scatterplots. Therefore, the following two sections provide the reader with a brief introduction to these two plots.

### 2.1.5 Scatterplot

A scatterplot is a commonly used plot in visualization research for conveying new ideas. The reason is probably because both experienced and novice users can easily decode and quickly understand the data items, which are commonly encoded as scattered points and displayed using a Cartesian coordinate system. A scatterplot uses two data axes, one horizontal (x-axis) and one vertical (y-axis), to represent two selected data variables. Each item in a scatterplot is represented by a 2-D point, or some other marker such as thumbnail or glyph, depending on its x- and y-axis values. A 2-D position is one of the preattentive attributes that humans perceive quantitatively by a high degree of precision, and scatterplots rely on this fact. A viewer can relatively easily and accurately using perceptual judgment compare quantitative values, judge various types of correlations between the used data, and do trend analysis, hence the frequent availability of scatterplots in visual analysis applications. For instance, the statistical technique of regression uses scatterplots to visually determine linearity. Linearity is one of the essential assumptions about data sets. Additionally, it is valuable for identifying data quality issues that may influence the precision of trained machine learning models. Scatterplots



are also effective for providing overviews and characterizing distributions. This plot is especially known in the interactive visualization community, which has contributed to many interaction techniques developed for scatterplots. The state-of-the-art interaction techniques are discussed in Section 2.2).

We have already presented two examples of scatterplots with different datasets. Figure 1.1 displays weather data, while Figure 2.2 depicts Anscombe’s Quartet. The scatterplot on the left in Figure 1.1 shows a rectangular brush, the most commonly implemented brushing technique in a scatterplot. The rectangular brush demarcates intervals on the two axes, and it does not change its size or shape when moved. Knowing the range of values on both axes selected by the brush rectangle makes it easier for the user to understand the data points selected by the brush quantitatively. However, a brush whose shape follows the coordinate axes’ orientation is not always beneficial. For instance, consider a scenario where the orientation of the elongated group of points does not align with the coordinate axes. In this case, the user might encounter difficulty—by using the above described rectangular brush—in exclusively selecting the points within the elongated and rotated cluster, i.e., without including other unwanted points in the brush’s rectangle. In this thesis, we introduce data-aware brushing for scatterplots, i.e., a brush that automatically changes its shape based on the underlying data structure, thus helping users to make more accurate selections.

### 2.1.6 Parallel Coordinates Plot

The parallel coordinates plot is one of the few plots that can effectively visualize up to a dozen dimensions of data. Although they have been well studied for many years already [Ins85, ID90, Ins09], and researchers and analysts often use parallel coordinates in information visualization, this plot is still less known among average users and rarely seen in presentations and newspapers. Because parallel coordinates do not need additional data transformations for showing the data, they are intuitive and easy to understand [SR06].

The axes of parallel coordinates are typically positioned vertically and equally spaced, hence the name of the parallel coordinates plot. Each axis in the parallel coordinates represents one data dimension. Figure 2.5 explains the idea behind the visualization design of the parallel coordinates plot, which is quite simple. As an example, we want to visualize five columns from the data table (a) that consist of 16 columns and 303 rows (selected columns are highlighted with a blue rectangle). The table in (b) displays only columns selected for visualization. The values in the raw data columns are usually not sorted, but for a better representation and a visual understanding of the correlations between different data dimensions, we will sort the columns as shown in (c) and then map them to the axes of the parallel coordinates. Because the columns are sorted, the lowest values appear at the bottom and the highest values at the top of each axis. Depending on the data values in a particular column, each axis can show a different range. Assuming that each row in the data table is one data object and the columns are different attributes (data dimensions) of that object, parallel coordinates will allow us to visually compare different attributes of one object or visually compare different attributes for multiple

objects simultaneously. For visualization, each object in the data set is mapped as a series of points, one per parallel coordinate axis, and polylines are drawn that intersect each axis to connect all data points. The position of each data point on an axis depends on the value of that data point in the associated data dimension. Figure 2.5(d) shows the final parallel coordinates plot. Data dimensions are assigned one-to-one to an equivalent number of axes laid out in parallel.

The parallel coordinates plot can help users easily find the range of individual attributes, outliers and clusters, and check for correlations between attributes. If the lines between the two neighboring axes are mostly parallel, they are correlated. Negatively correlated neighboring dimensions are characterized by lines crossing at a point between both axes. Moreover, users can observe a multivariate profile of each data object and so characterize data objects based on their multiple attributes simultaneously. However, understanding multivariate complexity using parallel coordinates commonly takes some time, and practice, especially if thousands of data (objects) items are displayed. Therefore, many extensions are added to the basic implementation of parallel coordinates, such as effective real-time dimension management and automatic clustering. Clustering techniques help reduce the number of shown data items in the case of overplotting [PWR04]. A re-ordering of axes allows users to arrange visualized data dimensions as they see fit—using additional meta information and heuristics is one way to find a good ordering [BTK11, XS20]. For a broader list of extensions, as well as their user-centered evaluation, see the survey by Johansson and Forsell [JF16].

The viewer sees which points on different axes belong together; in other words, similar objects are shown as having similar polylines. Additionally, we created a brush on the AvgTemp data axis to specify an explicit focus. Brushed lines are highlighted in color, while the rest is grayed out for context. Brushing could help us, for example, to easier find a characteristic multivariate profile associated with a higher average temperature, based on the other attributes shown. If the lines between the two neighboring axes are mostly parallel, they are correlated as are some of the brushed lines in Figure 2.5(d) between axes AvgTemp and MinTemp. We give an overview of brushing techniques for parallel coordinates in Section 2.2.

## 2. SCIENTIFIC CONTEXT AND RELATED WORK

(a) Raw Data Table: (16 Columns, 303 Rows)

(b) Selected for visualization: (5 Columns, 303 Rows)

Elevation	MinTemp	AvgTemp	MinPrec	AvgPrec
4195	33	49.4	0.37	1.32
1735	54.1	63.4	0.16	1.52
4400	30.4	46.9	0.3	1.01
335	56.9	65	0.01	0.94
1715	45.7	57.4	0.05	3.39
60	45.7	60.8	0.03	1.11
1708	47.1	62.9	0.1	2.21
2090	46.7	62.4	0.05	2.18
1292	45.9	60.5	0.14	3.04
25	57	62.7	0	0.99
940	46.8	69.3	0.07	0.31
489	47.2	65	0	0.54
1720	44.9	61.6	0.05	2.58
2320	47.4	65.7	0.05	0.36
2600	51.8	63.3	0.21	1.61
420	48.2	58.2	0.11	3.97
310	50	57.6	0.07	2.12
1250	41.1	57	0.21	3.16
6790	33.9	47.1	0.18	1.76
4102	38	56.2	0.13	0.42
5280	38.8	50.7	0.37	5.53
268	52.9	72.3	0.03	0.33
395	53.5	73	0.01	0.33
5575	26.3	42.8	0.5	1.86
8370	22.9	38	0.53	1.08
805	56.1	72.6	0.02	0.58
..	..	..	..	..
..	..	..	..	..

(c) Sorted Columns

Elevation	MinTemp	AvgTemp	MinPrec	AvgPrec
-194	22.9	38	0	0.19
-180	24.3	40.5	0	0.25
-112	26.3	41	0	0.25
-100	26.3	41.8	0	0.26
-64	26.6	41.9	0	0.26
-30	27	42.4	0	0.28
-21	27.2	42.8	0	0.29
5	28	44	0	0.31
8	28.6	44.1	0	0.32
8	29	44.4	0.01	0.33
9	29.2	44.5	0.01	0.33
10	29.2	44.6	0.01	0.34
10	29.5	44.8	0.01	0.35
10	29.6	45.2	0.01	0.36
10	29.7	45.8	0.01	0.37
12	30	45.9	0.01	0.38
13	30.1	46.2	0.01	0.38
14	30.3	46.4	0.01	0.39
..	..	..	..	..
..	..	..	..	..

(d) Parallel Coordinates (User Interaction)

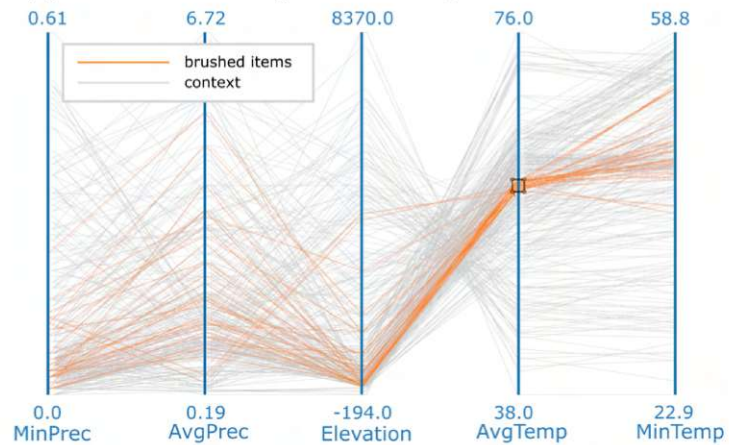


Figure 2.5: Parallel coordinates are used to visualize five out of 16 data dimensions of the meteorological data set California [NOA14]. (a) Screenshot of a data table displaying the complete data set (scaled down to fit in the view). (b) For visualization using parallel coordinates, the user can select about a dozen of columns (data dimensions); in this case it is five: Elevation, MinTemp, AvgTemp, MinPrec, and AvgPrec. All rows for the selected columns are included in the selection. (c) The values of each selected data dimension are automatically sorted before being mapped to the corresponding parallel coordinate axis. (d) Parallel coordinates plot with a rectangular brush created for selecting AvgTemp.



### 2.1.7 Coordinated Multiple Views

So far, we have emphasized how visualization can boost human visual performance, allowing users to perceive data patterns more effectively. A single visualization is sometimes just not enough to explain all the facets of the data, especially when dealing with complex and multivariate data. Two main approaches to show more views at the same time to the user are (1) juxtaposing them side by side and (2) superimposing the views as layers on top of each other. In the following, we discuss option (1), which integrates perfectly into the well-established concepts of interactive visual analysis (IVA). We will use the term *coordinated multiple views* (CMV) for juxtaposed views that provide interaction and are linked to each other. The following terms are also interchangeably used for the same fundamental idea: linked views, multiple views, coordinated views, and coupled views. Also, it is often the case in which when people mention dashboards, they implicitly mean that they have a special implementation of coordinated multiple views. Not all dashboards are interactive and linked and so do not fall into the CMV category. Recent research work by Sarikaya et al. [SCB<sup>+</sup>19] addresses the question of what dashboards are and how they are used.

The basic concept of CMV has developed rapidly from the initial idea implicit in *Brushing scatterplots* (Becker and Cleveland [BC87]). This idea involves cross-referencing corresponding data items in matrices of linked scatterplots. This development has evolved into the more general CMV concept proposed by Roberts [Rob07]. It is based on the idea that users understand their data better when interacting with the presented information and viewing it through different representations. In CMV, various graphics panels are appropriately linked, and selections of the data can be made in any of the linked views, using the technique nowadays commonly called linking&brushing. Hearst describes linking&brushing in CMV as “the connecting of two or more views of the same data, such that a change to the representation in one view affects the representation in the other views as well” [Hea99]. His definition is broad, given intentionally to emphasize one of the advantages of the linking&brushing concept, and that is, there are no strict rules regarding how to incorporate this technique into CMV, thus allowing for adaptation for specific use cases. The concept of linking&brushing has become a prevalent technique for data exploration and analysis and is key to IVA, as noted by Weber and Hauser [WH14]. Most research and commercial visualization tools, which support IVA, are centered around linking&brushing in CMV; examples include ComVis [MFGH08], and Tableau [Tab20]. As observed by North and Shneiderman [NS97], the CMV paradigm provides a unification of views and facilitates the recognition of previously hidden relationships within the analyzed data. Figure 1.1 is an example of using four different views combined in the CMV setup to help analyze weather data.

The widely known visual information seeking mantra [Shn96] introduced by Shneiderman outlines the most fundamental elements of interacting with the data shown in visualizations: overview first, zoom and filter, then details-on-demand. Following Shneiderman’s mantra, a CMV setup is often configured to show the overview first. However, this is not the strict rule, and depending on the task at hand, the CMV setup can also be

configured to start by exposing the most exciting details in the data. In any case, the idea of using CMV is to help users develop a comprehensive (mental) image of the analyzed data, especially if data relations are challenging to show in just one view. In this context, the main goal of visualization designers is undoubtedly to involve the user in a more profound analysis supported by the visualization of different data dimensions at once. While the user interactively selects new features for iterative drill-down exploration, the CMV system provides tight feedback in the form of computation/visualization updates.

With recent technological advances that have increased processor speeds and improved the design of traditional displays and 3D displays, new emerging fields of data exploration and analysis are opening up. Now, even large amounts of data can be successfully shown using CMV at large displays for multiple users [LKD19], or using carefully designed CMV visualizations on tablets [SS16], and even in mixed reality CMV environments [SKGM21, RBR22]. In general, selecting the “right” method for data visualization depends, on the one hand, on the data itself and, on the other hand, on the task that the user would like to achieve. However, to cope with more significant volumes of data in less time and more efficiently, it is often necessary to combine interactive visualizations used in CMV with computational analysis techniques. This approach is characteristic of visual analytics, which we briefly discuss below.

### 2.1.8 Visual Analytics

Research in visualization is primarily driven by users’ needs, i.e., aimed at solving specific problems, which users have when working with data. That is also how a prominent research discipline, visual analytics, started at the beginning of this millennium. It resulted from the need for new ways of solving certain problems whose sizes and complexities required a close connection between the human analyst and analytical methods. Data that visual analytics deal with are characterized by more than just the standard three attributes: volume, velocity, and variety. It also takes into account attributes like veracity, which characterizes the noise in data, and value, which offers insights into the associated costs and benefits. For a deeper exploration of the attributes of big data, please refer to the work by Kitchin and McArdle [KM16]. Because the sheer increase in the amount and complexity of data, the knowledge or value we want to extract can get buried in the data and is quite difficult to get to. Traditionally employed conventional methods for data analysis, such as automatic analysis, work reliably for well-specified problems, but are hardly effective enough to extract valuable information buried in complex and big data [TLCV15]. Visual analytics aims at reducing the time needed for the analysis, with the ultimate goal to discover knowledge and to make the right information available at the right time [BSS<sup>+</sup>19]. In their seminal work, Keim et al. [KKEM10] investigated challenges and opportunities concerning visual analytics research that lie ahead for several specific communities, including medicine, physics, astronomy, data management, and data mining. They concluded that visual analytics tools can help users to: synthesise information and derive insight from massive, dynamic, ambiguous, and often conflicting data; detect the expected and discover the unexpected; provide timely, defensible, and

understandable assessments; and communicate assessment effectively for action.

Although visual analytics has gained wide popularity due to its ability to solve challenging problems, its techniques are effective and efficient in various daily work processes. Visual analytics is used, for example, in the analysis of complex physical systems [MGH18], neurobiological data [GSF<sup>+</sup>19], public transportation [SDE<sup>+</sup>16], climate research and natural disaster management [CBKK<sup>+</sup>19, VSOC21], making statistical models more accessible for domain experts [M<sup>+</sup>18], and collecting and analyzing personal data [HTA<sup>+</sup>15, MGWM21]. All the examples demonstrate that to make the best possible use of massive

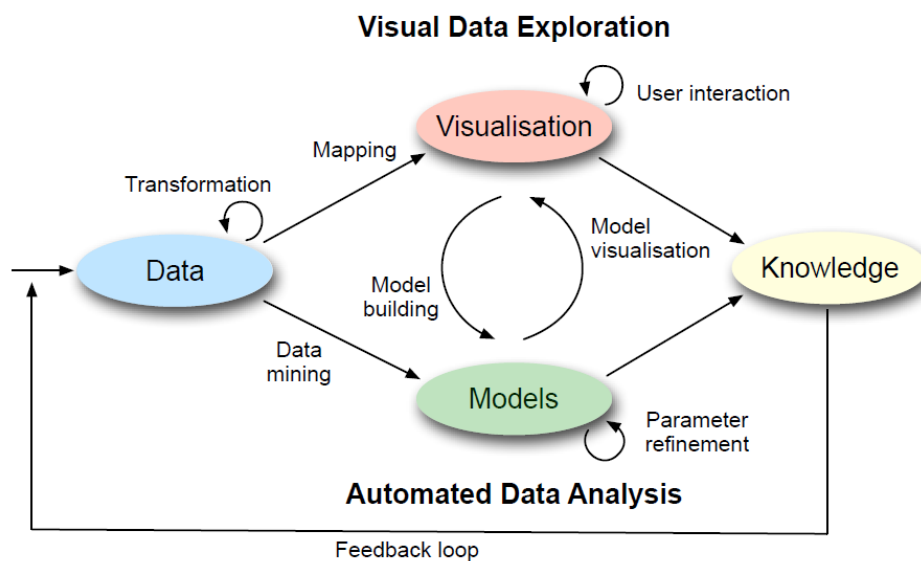


Figure 2.6: A process-oriented view of visual analytics [KKEM10] is characterized through interaction between data, visualizations, models about the data, and the users in order to synthesize knowledge from data.

data and information computed, acquired, and stored by modern analytical systems, it is increasingly imperative to actively involve human intelligence in visual analytics processes. The term *visual analytics* was coined by Thomas and Cook in the research and development agenda *Illuminating the Path* [TC05], where they also defined visual analytics as the science of analytical reasoning facilitated by interactive visual interfaces. As Keim et al. [KMS<sup>+</sup>08] explain, visual analytics tightly integrates computational tools, visualizations, and interactive methods to utilize the visual perception and analysis capabilities of the user. The book *Mastering the Information Age: Solving Problems with Visual Analytics* [KKEM10] discusses the role and importance of including the user in visual analytics processes. Moreover, the authors of the book emphasize the vital role of a feedback loop (see Figure 2.6) that makes it possible for the user to continuously refine his findings at different stages of an analytical process. Visual analytics, a multidisciplinary field, owes its agility and rapid progress to the integration of techniques from diverse

research areas. Bringing various areas together has a positive impact because progress in one promptly stimulates and extends progress in another. Advances in any of these areas directly increase the power of visual analytics. In terms of interaction methods, visual analytics not only discusses interactions with graphical elements that represent data but also more profound concepts about interacting with data itself, which are very much needed in analytical processes. For this master thesis, the most relevant parts of the visual analytics process are the visualization stage and interaction methods. The visualization stage serves as the medium, and the interaction methods means to involve the users' judgment in the process. In fact, when we support data visualization with techniques that allow the user to interact directly with the displayed data, we realize one of the most successful visual analytics concepts, namely, *interactive visual analysis*. We delve into this concept in the following section.

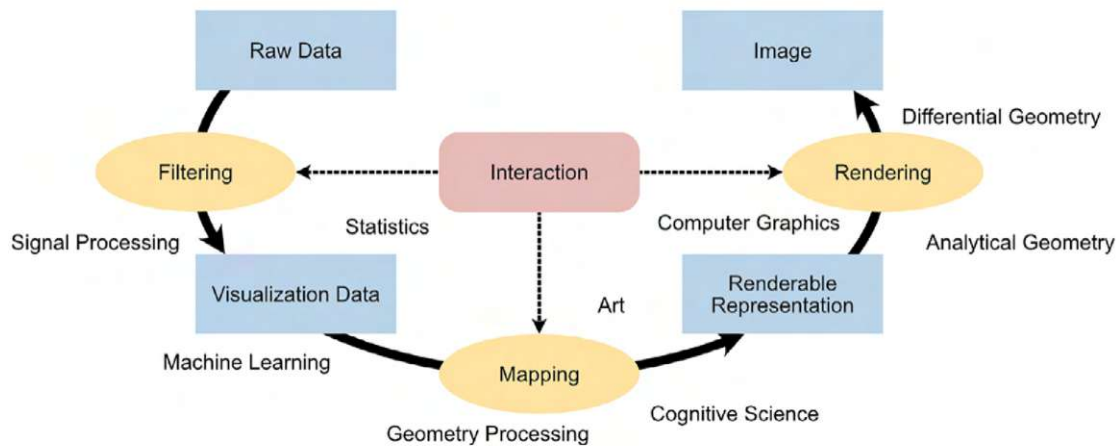


Figure 2.7: Classical visualization pipeline [HM] annotated by Reina et al. [RCM<sup>+</sup>20] to display examples of influences from other disciplines at its different stages.

## 2.2 Interactive Visual Analysis

An important milestone in interactive visual analysis (IVA) was created by McCormick et al. [MDB87] in 1987, who emphasized the fundamental importance of graphical representations and interactive manipulations in science and technology. Instead of dealing with a great quantity of numbers, the analyst should be able to perform an action on the graphically represented data through dynamic graphical methods, by utilizing the great bandwidth of the human visual system for encoding visualizations. Cleveland and McGill [CM88] also noted that for various analytical tasks, including exploratory data analysis and process monitoring, providing interaction is one of the most important parts of the design process, which brings us beyond the limitations of static visualizations. The advantage is that users can modify their visualizations and then observe the changes to

understand their data better, and gain potentially useful information hidden in the data.

Compared to modern visualizations, early interactive visualizations offered only the most basic interactions, such as focus+context [Hau05] visualizations. This feature enables the user to zoom in on specific regions or particular data subsets of interest, and zoom out to view the entire dataset, providing both a detail and context overview, i.e., the big picture. Nowadays, to assist users in addressing each emerging question, IVA enables them to adjust what they see and how they see it. Users can control the stages of the visualization pipeline, such as governing the generation of the visual representation of data and filtering or manipulating data as needed to learn from its diverse aspects. Figure 2.7 by Reina et al. [RCM<sup>+</sup>20] reveals the central role of interaction, it appears at many places. Interaction enables the user to fine-tune a large number of parameters that are usually left to be adjusted to steer the pipeline execution in the wanted direction. Although the stages of the visualization pipeline shown in Figure 2.7 are presented as a sequence, the IVA process is not sequential and is highly iterative. The work by Reina et al. [RCM<sup>+</sup>20] is further interesting because it gives a comprehensive view on the field of visualization research. It includes the discussion of identified main challenges related to the production of novel visualization approaches, with a reflection on the user experience and underestimated complexity of the ecosystem beyond the visualization process. The authors comment that implementing new methods based on external technologies, such as statistical modeling and deep learning, can open up promising research avenues and inspire novel approaches to old problems [RCM<sup>+</sup>20]. For those interested in fundamental components of the visualization process, from the data to the human viewer, we recommend reading the updated edition of the well-known book by Ward et al. [WGK21] titled *Interactive Data Visualization: Foundations, Techniques, and Applications*. In the following, we summarize the previous research on interaction techniques and give an overview of IVA technologies related to this master thesis.

### 2.2.1 Interaction Techniques

During the long time that interaction has been an integral part of data exploration and analysis research, various visual analysis techniques have been presented to help analysts amplify thoughts, specify their focus, and steer computations. In the early 1970s, John Tukey's team developed PRIM-9 [FFT75], one of the first systems for manipulating multidimensional data. At that time, it facilitated direct interaction with graphical elements on a computer graphics screen, accommodating up to nine dimensions. PRIM-9 offered a set of dynamic tools, such as isolation (a precursor to brushing), projection, rotation, and masking. Using PRIM-9, users were allowed to manipulate the visualized data, and the system would instantly adapt by displaying pertinent changes on the screen. Focus on subsets of data items was possible that might otherwise be overlooked when using computational methods alone. This presented an entirely novel experience for users who were accustomed to static visualizations.

In the mid-1980s, when computers became more affordable and their power increased, the field of visualization experienced a rapid development. Becker and Cleveland in

1987 [BC87] defined the term *brushing*. They used brushing as an interactive method for highlighting, shadow highlighting, deleting, and labeling groups of data items in real-time on a scatterplot matrix. A scatterplot matrix is a rectangular array of all pairwise scatterplots of the variables, and they are *linked* together so that the effect of a brushing operation appears simultaneously on all scatterplots. The concept of brushing in one view and updating in other linked views later became known as *linking&brushing*, a general approach to interactive visual exploration and analysis of multidimensional data using coordinated and multiple views (CMV) [Rob07]. Linking&brushing supports, for example, understanding of correlations across multiple data dimensions [BC87, BMMS91].

Brushing is an established technique and the first type of interaction considered when designing an interactive visualization. It also serves as an enabling mechanism for other essential techniques in IVA, such as focus+context visualization [Fur86, CMS99, Hau05]. By definition of focus+context visualization, subsets of data items that are selected by brushes are visually discriminated from their context, i.e., the rest of data items. That can be achieved, for example, through the uneven allocation of graphic resources for visualization, including space, opacity, and color. In combination with linking&brushing in CMV, this results in a particular visual experience for the user. When the user initiates a brushing operation in the brushed view to focus on specific data items of interest, two things occur. First, the selected data subset is immediately brought into focus. This might involve changing the color of the brushed data items. Meanwhile, all other data items are designated as context. This might entail displaying them in a gray-scale style. Second, linking is promptly established in all CMV views. This is done consistently, such as using the same coloring across all the linked views. One example of linking&brushing in CMV is already shown in Figure 1.1.

The basic principle of brushing involves interactively drawing a shape (a bounded region) around interesting data items. For example, the circular brush is anchored in its center, and it selects all data items inside as defined by the radius. Although many different brush shapes for selecting a data subset of interest were proposed—including squares, circles, and polygons, to name a few [BC87, CM88, War94]—visualizations typically provide just one predefined shape, commonly a rectangle. The shape of the brush must be carefully selected to comply with the visualization properties [Wil05]. For instance, a parallel coordinates plot mostly uses a rectangular shape for a brush placed over a single axis. The brush can change its size only in one direction, i.e., along the axis. In a 2-D scatterplot, both circular and rectangular brushes work well with two orthogonal axes, but this is particularly true when the aspect ratio is 1. These brushes are very sensitive to the aspect ratio, so the brush becomes stretched if only one side of the view is resized. In such case, the user might find it challenging to understand the brush quantitatively. In addition to the rectangular and circular brushes, many other brushing techniques have been presented for a scatterplot and parallel coordinates, which we discuss below.

Some data analysis tools provide a simple square brush as a default, but allow the user to alter corner vertices of the square to create a quadrilateral of a different shape and size. Unique shapes are possible using the lasso brush, which originates from image editing



tools, where the lasso tool provides a way to draw a bounded region of any shape and size. This approach is very accurate—the user creates a unique brush by clicking and dragging to draw a line directly below the current mouse position until the outline of the desired shape is generated. The lasso brush has one major drawback in the IVA context: it does not work fast. Designers of interaction techniques for IVA need to be careful concerning the time the user needs to create a brush. The more the user focuses on creating a brush, the more he deviates from the task he started. That is, he leaves the flow of data analysis he started. According to Card et al. [CRM91], if the user has to perform more than very few atomic interactions, like clicks, to create a brush, that may interrupt the analysis process.

In general, analysis tools provide a predefined set of interaction techniques, and the user is mostly left without possibilities to define their own actions in brushing and linking interactions. One possible reason for customization could arise from the specific requirement to make data selections in a way that isn't supported by the default settings. Also, it was found that users often desire to feel that they are in control over the system's actions [CMK<sup>+</sup>12]. One of the first attempts to help users create their own custom settings in the context of brushing was made by Koytek et al. [KPV<sup>+</sup>18] who integrated several linking&brushing techniques to create an interface called MyBrush. They aimed to augment linking&brushing interactions by incorporating personal agency, which offers users the flexibility to configure the source (what is being brushed), link (the expression of the relationship between source and target), and target (what is revealed as related to the source) of multiple brushes. The most significant drawbacks here are that the configurability of MyBrush depends on the number of available techniques and that the whole process is time-consuming. For example, the user is responsible for defining which views are linked to the brush. Although these settings take much time, they ultimately give the user the power to decide how and which linked views will highlight the selected data.

Further notable variations of brushing techniques include: 1) *smooth brushing* [DH01] for fuzzy selection of data items. This brush is useful for data dimensions with gradual changes, which are often found in scientific simulations. 2) *angular brushing* [HLD02] that works in parallel coordinates for emphasizing rational data-properties, i.e., features which depend on two data dimensions, instead of one. 3) *selective angular brushing* [SGMS21] that works in parallel coordinates for a single-click selecting of all lines that follow a certain angle, starting from a point or range on an axis. 4) *line brushing* [KMG<sup>+</sup>06] in curve view that provide a fast way, for example, to exclude outliers in a family of function graphs. 5) *N-dimensional brushing* [War94] that enables creation of brushes of the same dimensionality as the attribute space. 6) *sketch-based brushing* [FH18] that works in a scatterplot and enables the user to effectively select specific data subsets, even when their geometric delimitation is non-trivial.

The concept of composite brushes is also very useful [MW95]. It allows for configuring a composite brush which includes more than one brush. The user applies logical operations and expressions, including AND, OR, XOR, and NOT. If we use CMVs to assemble a

complex brush, we are not limited to brushes within a single view, but we can create additional brushes in any linked view, and so we can create very complex queries.

Doleisch et al. [DGH03] introduced a feature definition language for the specification of multidimensional and/or complex features, using logical combinations of brushes in CMVs. Splechtna et al. [SBG<sup>+</sup>18] introduced cross-table linking and brushing for interactive visual analysis of multiple tabular data sets without a unique key. Their approach is based on establishing single-directional or multi-directional data table links and an implicit brush that has no explicit visual representation but still acts as a regular brush in the sense that the selected items are highlighted in all linked views. A specially designed user interface is required to keep track of brushes, links, and back-links, i.e., the inverse mappings used to determine the refined brush in the original data set.

Brushing with the support of machine learning has recently become more popular. We have already mentioned sketch-based brushing [FH18], which also exploits a convolutional neural network (CNN) for estimating the intended data selection from a fast and simple click-and-drag interaction and the data distribution in the visualization. Another interesting approach is demonstrated by Gadhav et al. [GGC<sup>+</sup>21], which, instead of creating data-aware selections, actually algorithmically derives the underlying pattern of a selection. The algorithm aims to determine the common characteristics that group the items within a selection together, as well as what distinguishes them from other data points.

Konyha et al. [KLM<sup>+</sup>12] found that IVA has different levels of complexity. At its basic level, IVA builds on the combination of different views and provides only one brush in a view for interactively selecting data items. Users are allowed to adjust their brush interactively, for example, to move and resize the brush, as needed to select what they find interesting. The second IVA level allows for complex, composite brushes. A composite brush is built using various logical combination schemes and it can consist of multiple brushes created in different views. At this level, the user is provided even more freedom in the sense that he can iteratively start a deeper information drill-down to answer complex questions about the data. At its third level, IVA combines complex interaction and general information extraction mechanisms. There exist two (partially complementary) approaches to extract deeply hidden implicit information from complex data sets: (i) first derivation of additional attribute(s) and then visualization and brushing, and (ii) usage of an advanced brush to select “hidden” relations in the data. The distinction lies in that approach (i) employs a simple brush on complex data, whereas approach (ii) involves creating an advanced brush on simpler data. During approach (i), the user needs to complete several steps during IVA. Since a simple brush is used, this approach commonly requires more viewing space. For example, several views may be necessary, as well as on-demand data computation, such as interactive attribute aggregation and derivation. These processes can be facilitated for the user during IVA with the help of advanced derivation dialogs [KLM<sup>+</sup>12]. The approach (ii) uses advanced brushing and can provide the same insight from the data using just a single step and one view. However, advanced brushes require complex interactions, and uninformed users must learn how



to use such brushes for an effective IVA. Ward et al. [WGK15] provides an overview on theory, techniques, and tools necessary to build systems involving the interactive visualization of data.

Brushes techniques are commonly categorized into three groups, according to the space in which the selection is performed: screen, data, and structure brushes [FWR00]. While brushing in screen-space traditionally limits the shape of a brush to two dimensions, brushing in data-space permits brushes with dimensionality greater than two. For example, the N-dimensional brush [War94] provides insight into a spatial relationship over N dimensions. The third group extends the brush metaphor to structures. Structural relationships between data items, such as clusterings, orderings, and groupings, can be considered by the brush [FWR00]. Brushing in structure-space is beneficial for data sets with natural and imposed structures. The Mahalanobis brush, which we explain in Section 3.6, is a new structure-based brushing technique.

Traditionally, brushing has been performed unconstrained, meaning that brushes can be created anywhere in the view, and the analyst can move or resize them freely. In addition to the free (unconstrained) brushing, and to support reproducibility, we introduce an alternative brushing that we call *constrained brushing*. Also, we extend the range of available interaction techniques in IVA by introducing two percentile brushes and the Mahalanobis brush. We do this by following the concept of brushing through direct manipulation of data items shown in a visualization. This is in line with the definition of focus+context visualization [Hau05], which describes brushing as an explicit, on the view focusing interaction.

## 2.3 Reproducibility within the Context of Visualization Research

Reproducibility is one of the essential characteristics of science, and it is widely recognized as a critical aspect of analytical ecosystems [Bak16]. By looking at things around us, systems that have become or will soon become part of our daily lives, we recognize the necessity towards technologies that demand reproducibility. A simple example is a fully-automatic coffee machine, where reproducibility guarantees that the system provides a reliable response, making a coffee of the same good taste we used to get every time we use it. Reaching a consensus on what reproducibility means is very important. The American National Academies of Sciences, Engineering, and Medicine (NASEM) has recently released the report named “Reproducibility and Replicability in Science” [oSEM19], which defines the terms reproducibility and replicability as applied to scientific and engineering research. According to the NASEM report, reproducibility means obtaining consistent computational results using the same input data, computational steps, methods, and conditions of analysis. Replicability is explained as obtaining consistent measurements or results using new data, methods, and/or conditions in a study aimed at the same or similar scientific questions.

It is not always possible to achieve complete reproducibility, for example, due to sensitive data that cannot be shared. Therefore, it is advisable to communicate such constraints, for example through the use of models such as PRIMAD [FFR16] that provides several pre-defined variables used to describe which aspects of the experiment can be changed while still attaining reproducible results.

The reproducibility challenges faced by the visualization community are recently discussed by Fekete and Freire [FF20]. The authors outline a set of recommendations for the different types of visualization research domains based on the findings and recommendations of the NASEM report and the research on this topic from the “EuroVis Workshop on Reproducibility, Verification, and Validation in Visualization” [Eur]. Also, it confirms our view on the need for reproducibility of brushing operations performed in interactive visualization systems.

### 2.3.1 Provenance Tracking

In computational and data science, reproducibility and replicability problems are often tackled by means of literate programming—Donald E. Knuth coined the term literate programming in 1984 [Knu84]. It represents a programming paradigm that aims at explaining the “how and why by any means necessary”, which includes writing clean and readable code, adding markdown text, images/visualizations, equations, videos, and links that describe what is being done at each step of the analysis. Literate programming is supported by many computer programming environments, including Jupyter Notebooks [Not], Observable [Obs], and R Markdown [RMa]. These environments combine code and explanations, i.e., required aspects of literate programming, to support storytelling and, more importantly, to assist others in reproducing a stored analysis. For example, the LIGO open science center that detected gravitational waves in late 2015 has put some of their research results in a notebook form, together with all required data files [LIG]. Other analysts around the globe can anytime replicate a signal processing described in that notebook, and even do some additional analysis based on the previous results. Creating a good notebook that is carefully curated and well narrated is a time-consuming multistage process. Moreover, processes in interactive visual analysis that involve the user’s interaction with the visualized data are often complex and require many trial-and-error steps. They are very challenging to track and make reproducible using standard environments that support literate programming.

Many frameworks designed for recording user interactions with visualizations have already been introduced, such as EvalBench [AHR13] and GraphUnit [OJ15]. Most commonly, the primary purpose of these frameworks is logging, for example, for conducting user studies for better understanding of a user’s interaction with the visualization system, with the aim to support or improve usability and performance of the system. The recorded steps can be repeated, but not reproduced since they involve a variable that can not be controlled, and that is human variability. An overview of the measures to account for human variability in the context of reproducibility in visual analysis is given by Fekete and Freire [FF20]. There is no general rule to record user interactions. The main concern

in making a helpful history, for example, when interactions such as moving a brush in visualization are included, is determining a reasonable granularity of changes that should capture user’s intention. A direct approach for capturing a user’s interaction is logging low-level information, including the keystrokes, the (x, y) positions of the mouse (as used in [GL12]), and a mathematical description of a brush, such as the center point and radius in the case of a circular brush. In most cases, only interactions at a higher semantic level are saved that reflect a specific action defined by the visualization tool itself. Such an approach was used by Battle and Heer [BH19] to identify repetitive patterns across participants’ exploratory visual analysis using a popular visual analysis tool named Tableau. They captured only high-level actions such as “shelf-add,” and “shelf-remove,” which are Tableau-specific interaction elements. From a technical point of view, it is possible to design tools that allow the interaction to be captured at any level of detail, but there are many aspects to keep in mind, such as ensuring efficient storage. For example, consider saving each interaction over long periods of time such as storing mouse coordinates as a sequence by adding new information at every position change, even if the user moves the mouse over parts of the plot where there is no data under the mouse. This can lead to huge log files that support reproducibility, but are not appropriate for a higher-level analysis. However, omitting details can lead to loss of information—intermediate steps taken by the user can be vital to understanding his intentions and decisions made along the way [GW09, NXW<sup>+</sup>16].

The visualization community is conducting significant research on history recording, better known as *provenance tracking*, which is recognized as a vital feature of visualization systems that, among other benefits, support collaboration and reproducibility [DF08, KCD<sup>+</sup>09, Sta14, XOW<sup>+</sup>20]. The definition of provenance is “The place of origin or earliest known history of something” [198]. In the context of visualization, provenance tracking means a record or a log of everything that lead to the current visualization state. It involves the automatic recording of methods, actions, and parameters used in different stages of a tracked visualization process, such as during data collection, analysis, and visualization. Recorded provenance is then often analyzed, for example, to understand a user’s analysis behavior (to learn a model about the user) or to support sensemaking tasks of the user. A recent survey by Xu et al. [XOW<sup>+</sup>20] explored works done on analysis of user interactions and provenance data by structuring related work around three primary questions: (1) WHY analyze provenance data, (2) WHAT provenance data to encode and how to encode it, and (3) HOW to analyze provenance data. We refer to Ragan et al.’s survey [RESC16] for a more complete picture of different perspectives of provenance that are most relevant to the areas of visualization and data analysis.

Provenance tracking approaches are commonly divided into two groups [FKSS08]. The first group aims at tracking provenance by recording the analysis process, while the second group tracks provenance through an explicit workflow modeling system. The former approach is also known as *process-based* and the latter one as *workflow-based*. Both approaches have their advantages and shortcomings, which we will briefly explain by taking the well known VisTrails [SFSA10] tool and the Track [CGL20] library as

representatives of the two approaches.

VisTrails is an open-source scientific workflow management system (SWfMS) for data analysis and visualization that follows the approach of capturing provenance through an explicit workflow modeling system. For an overview of data-intensive SWfMSs see a survey done by Liu et al. [LPVM15]. VisTrail and similar SWfMSs, such as Kepler [LAB<sup>+</sup>06], and Taverna [OAF<sup>+</sup>04] facilitate scientific investigation and discovery by giving the user the possibility to add new modules and link them into the pipeline appropriately as needed. Reproducibility in VisTrails is supported by automatically keeping track of the changes made during development, such as the refinements of a computation method used or different visualization algorithms applied, and allowing the user to run preserved workflows to reproduce the results. For performance reasons, VisTrails uses *change-based provenance*, i.e., it captures only the actual actions the user employed to transform the workflow. Also, this enables action recovery (undo/redo), and helps the user to follow the evolution of a solution, visualized as a series of activities using a directed acrylic graph (DAG), and to reproduce the steps (nodes of a DAG) used in building the visualization pipeline. One of the disadvantages of VisTrails is that while it supports collaboration through sharing workflows, it does not support real-time collaboration between several users of the same workflow. A major challenge in supporting multiple users working concurrently on the same workflow is providing consistency in the event of conflicting concurrent operations. Mostaeen et al. [MRRS18] proposed a locking scheme that supports locking workflow components at a granular level in addition to supporting locks on a targeted part of the collaborative workflow. According to the authors, their technique can reduce the average waiting time of a collaborator by up to 36.19% in comparison to existing descendent modular level locking techniques. For an example see work by Zhang et al. [ZKL14]. SWfMSs that implement an explicit workflow modeling approach primarily focus on tracking and managing provenance information related to the construction of complex visualizations. Although this provenance tracking approach provides detailed provenance of exploratory computational tasks to simplify the process of exploring data through visualizations, to the best of our knowledge, current solutions do not provide support for the reproducibility of a brushing operation itself.

Another popular approach to capture provenance data is process-based, i.e., it records the analysis process as a sequence of actions in an interactive system. One advantage of loosening the requirement for explicit workflow modeling and moving to build a provenance graph by adding a new node in the graph for each recorded user action, is easier integration into existing and future visualization systems. The system itself must be state-based, e.g., it should keep track of the state of interaction). Examples include reproducible tracking (Ttrack) [CGL20], shareable interactive manipulation provenance (SIMProv) [CCSK19], graphical histories by Heer et al. [HMSA08], and the scripting-driven propagation system (CzSaw) [KCD<sup>+</sup>09]. Since Ttrack is one of the latest developments, we briefly discuss its characteristics.

Ttrack is a web-based library for provenance-tracking in web-based visualizations. It can help visualization system designers to easily incorporate a state-tracking mechanism

that maintains changes made to the visualization, including configurations, filter settings, and user selections/annotations. Ttrack enables a recall of the analysis process by visualizing the provenance information and the reproducibility of any of the recorded states. The recordings represent persistent action, such as adding a plot or making a brush). Developers must define a state that fully describes their application. Each user interaction that should be captured must be created and applied when a user interacts with the visualization. For faster switching between different states, especially between actions that are distant from each other on the provenance graph, Ttrack implements a special model called the differential states storage model. As the authors of Ttrack note, due to the highly dynamic nature of the visual analysis, additional investigation is needed to optimize strategies for storing states, primarily by finding the optimal frequency of storing new states and deciding on ideal times to store states. The Ttrack library was used recently by Gadhav et al. [GGC<sup>+</sup>21] to implement a prototype system that uses provenance data for detecting, predicting, and ranking *pattern-based intents* behind a current selection in a scatterplot. A pattern-based intent is an intent of the user, for example, to select a cluster or an outlier. Such a user reason to perform brushing can be automatically captured and added to the provenance graph. This is in contrast to domain-specific intents, for example, reasons for starting analysis, that are commonly tracked by annotating them.

For demonstration, Gadhav et al. [GGC<sup>+</sup>21] implemented a set of selected methods in a scatterplot that is useful for automatic completion and tracking the provenance of pattern-based intents. Their approach supports recall and reproducibility by explicitly tracking the intents and their constituting interactions. The system responds to any change in a data selection by showing ranked predictions of patterns for a selection, for example, after changing the extent of the brush. It is up to the analysts to capture their pattern-based brushing intent by verifying a prediction.

In our work, we also consider different patterns and ranks that help users make meaningful selections and/or data-driven decisions. In our approach, we propose to design a structured brushing space that influences how users interact with the data by semi-constraining the brushing operation itself. One of the advantages to use constrained brushes is that they are fast. Users can easily understand what is selected with a constrained brush and quantitatively interpret the selection. As our investigation of the state-of-the-art shows, the visualization community is still researching strategies and methods for capturing and preserving the provenance of brushing operations.

### 2.3.2 Animation supports Reproducibility

Given the range of concepts for which animation seems appropriate, its widespread use in visualization was to be expected. Animations involve showing the viewer a series of frames in a defined sequence. In addition to the benefits we outline below for visualization research, the animation framework itself offers the capability to reproduce animated frames. This inspired us to consider introducing the animated brushing, which in turn supports the reproducibility of the brushing operation.

As early as 1975, PRIM-9, one of the first interactive systems, was enhanced with animations to simulate the automatic rotation of the visualized data around a selected axis of rotation [TM87]. The new feature enabled the acquisition of additional insights into the data by observing the animated changes in the visualized data that come in response to a change in the observer's point of view.

Animation is a well-known technique for enhancing (interactive) presentations. We mentioned the well-known Gapminder Trendalyzer tool and one of the recently presented examples of using this tool to summarize the progress of sustainable development goals of the United Nations [Fou20]. Modern commercial visualization tools like Tableau [Tab20] have also added support for animations in the visualization. Animated transitions can reduce errors regarding the estimation of changes in the data [HR07], and support object tracking and building mental maps of spatial information [BB99]. In the same work, the authors also provide a taxonomy of animated-transition types and describe their differences. Tableau implemented all the guidelines for animated transitions given in the research paper [HR07]. Moreover, Tableau [Tab20] provides the following explanation as a motivation for users to create animated charts: "Without animation, changing something like a data filter causes scatterplot marks to suddenly jump to new locations. It's hard to pinpoint what changed or why, but a smooth animation connects the dots. It's easier to spot and understand changes, like when a specific mark becomes an outlier, when there's a sudden value spike or dip, or when data clusters appear. You can sense how bars grow, shrink, or re-sort relative to each other or track an individual mark's path".

In a more recent work [KCH19], the design and evaluation of animated transitions were investigated in the context of conveying aggregation operations in visualizations. Study results indicate that judiciously staged animated transitions can improve subjects' accuracy at identifying the aggregation performed. Another interesting recent study investigated the effect of animated transitions on data sets with missing data. The visualization of missing data is important [SS19], but the use of animation in this context has not been thoroughly studied. Even static views that display only a single time step, can be animated [LFW<sup>+</sup>20]. Wu et al. [WJXN16] used animated bars in their study to investigate the perceptual accuracy for animated data visualizations, both for presentations and as part of interactive applications. Animation is also very often used in visualization research. It serves as a means to maintain insight about visualized data during view and/or perspective changes. These changes can involve actions like hiding and revealing structure, switching between detail view and overview, and zooming in and out [SI08]. Animation is also employed if visual changes occur due to data manipulation by users, for example, in cases of streaming data [HVF13], multidimensional data [EDF08], dynamic networks [BPF14], and patient cohorts [RAM<sup>+</sup>11].

From the examples shown, it is evident that animation plays an important and a supportive role in visualization. With the help of animation, we can incorporate temporal information into different views, for example, to visualize how data changes over time. Or we create transitions and transformations that make static visualizations more compelling and effective. As various sources note, animation should be carefully utilized, as it is not



always beneficial. The apprehension principle, a fundamental guideline in information visualization and graphic design, states that graphics should be accurately perceived and appropriately conceived. Tversky and Morrison [TMB02] found that too complex or too fast animations can not be perceived accurately. Animations may violate the apprehension principle of good graphics. The same authors note that animations are suitable for conveying concepts of change, for example, for expressing processes such as weather patterns or real-time reorientations in time and space.

Tversky and Morrison conclude that although animations may be less intuitive in conveying complex systems, this aspect can be improved by adding interactivity. They found that adding interactivity judiciously can solve many animation problems, stimulate interest, and encourage users to explore visualizations. Other authors like Robertson [RFF<sup>+</sup>08], also note that animation must be used with caution since it could lead to perceptual errors and can slow down the analysis. As many data analysis cases to date have shown, carefully designed animation supports visual analysis. The design guidelines for animated visualizations are discussed by Danyel Fisher [Fis10]). Also, high-level languages are proposed that enable users to specify the animations for data charts (see Kim et al. [KH21], and Ge et al. [GZL<sup>+</sup>20]).

This thesis uses animation to ensure reproducibility of the brushing operation and to improve the user's understanding of the changes, associated with the brushing operation, in the linked views.

## 2.4 Enhancing Qualitative Analysis with Quantitative Information

Visualization pioneers recognized that humans process numbers serially, one by one, and that the visual representation of these numbers accelerates perception, inquiry, exploration, and understanding of the stories contained in the numbers. Hence the efforts to research and improve the ways to qualitatively present data in visualizations. The qualitative nature of the interactive visual analysis (IVA) results has been and still is one of the main reasons for its success. More recently, as the dissemination and complexity of the data to be analyzed has increased, visual analysis has begun to find applications in entirely different fields, from virus research in microbiology to space exploration in astronomy. In many application domains, such as business analysis and medicine, conclusions must be drawn on the results from data analysis. Where exact numbers and hard facts are needed, people refrain from drawing critical conclusions based on the results of a qualitative analysis.

In order to support the application of interactive visual analysis in a variety of cases, especially if decision-making is paramount, the visualization community needs to find appropriate ways to enrich the qualitative visual analysis with additional quantitative information. The first steps towards this goal have already been achieved, and we



briefly review the work on improving the visual analysis through additional quantitative representations.

Since Anscombe [Ans73] demonstrated the importance of graphical representation for a better understanding of numerical values in raw data tables, his example has been traditionally used to emphasize the advantage of graphical data visualization over tabular data representations supported by basic summary statistics. Humans are not capable of gaining insight into data relations just from looking at tables containing thousands of numbers, and sometimes even dozens of numbers can be a challenge. Visualization alone, like in Anscombe’s demonstration, is not enough to provide the correct understanding of the statistical relationships between the four data sets he uses. By looking at the four scatterplots shown in Figure 2.2, the viewer will probably need to think carefully to understand that the basic statistical profile of all four data sets is very similar. For example, the outlier point situated in the upper-right corner of the bottom-right scatterplot is enough to produce a correlation coefficient of 0.82 between  $x$  and  $y$ —the other three scatterplots also share the same correlation coefficient. This is a good example where we can combine the analysis capabilities of the human user with the strengths of automatic data analysis to speed up interactive data exploration and analysis. As descriptive statistics offer valuable quantitative readings, in this case, the most straightforward solution to support the user’s comprehension of the visualized data is to display these numbers overlaid on a scatterplot.

Graphical overlays are traditionally used to aid chart reading. They allow displaying of, for example, summary statistics, such as the mean, median, and standard deviation. Moreover, summarizations of the data are commonly used as representative information for clusters in hierarchically organized large data sets [Shn92, FWR00]. Summary statistics, when displayed to the user, can save the user from time-consuming cognitive loads during the mental calculation of aggregated statistics from data [TT06]. In this regard, the user can be further supported by displaying animated transitions that convey aggregation operations in the visualization, for example, starting from a subset of data points in a scatterplot and arriving at the resulting summary statistic value. For more details about animated transitions see the work by Kim et al. [KCH19].

Traditionally, interactive visual analysis is centered around the linking&brushing technique, which is modeled as an interactive and iterative method to reveal insight into large and complex data sets. If summary statistics are combined with brushing, it is helpful to keep track of the calculated statistics. To avoid generating a visual clutter in the brushed view, commonly, only the values for the current brush are displayed, and the past values are either saved in a table or showed in another view [PCW89, MW95]. Other types of overlays, such as reference structures, highlights, annotations, interactive overlays, and layering, are discussed by Kong and Agrawala [KA12]. Our work contributes to reference structures with various grid types that can be placed over one data axis or as a 2-D overly in a scatterplot.

Instead of providing data statistics for all data, they can be calculated on demand. For example, Haslett et al. [HBC<sup>+</sup>91] showed the average of the points that are currently

selected by a brush. Kehrer et al. [KFH10] integrated statistical aggregates along selected, independent data dimensions in a framework of coordinated, multiple views. Brushing particular statistics, the analyst can investigate data characteristics such as trends and outliers. Chen [Che03] showed how to enable analytical filtering through the addition of the quantile range-filter for one variable to validate or filter data selections.

Our goal is to strengthen interactive visual analysis regarding the weaknesses mentioned above, i.e., lack of reproducibility and quantitative results. To support the quantitative analysis, in our work, we emphasize the importance of extensions that enable summaries from the brushed data and presentation of summary statistics in the linked views in the form of tables or overlays.



Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar  
The approved original version of this thesis is available in print at TU Wien Bibliothek.

# Reproducible Brushing

Reproducibility enables the verification of published findings and generally ensures that we can build on previous knowledge and research results achieved by our colleagues or ourselves. As discussed in Section 2.3, there is a growing number of studies related to reproducibility in the context of visual analysis. With our work, we contribute to the state-of-the-art by concentrating on interactive visual analysis (IVA), where we propose several techniques to improve the reproducibility of the brushing operation.

This chapter is structured as follows: In Section 3.1, we introduce the *Structured Brushing Space* and show two concrete implementation examples using the *Snap-to-Grid* option for brushing in Section 3.2 and the *Percentile Grid* in Section 3.3. Grid extensions for a parallel coordinates plot are discussed in Section 3.4. We then introduce two brushing techniques which enable new ways for rank-based analysis. These are the *Percentile Brush* in Section 3.5, and the *Mahalanobis Brush* in Section 3.6. Moreover, in Section 3.7, we describe the animated brushing technique as a general way to support the reproducibility of interactive visual analysis using various views and different brushing techniques.

## 3.1 Structured Brushing Space

*Brushing* is a well-known technique introduced many years ago [BC87] that users of IVA know very well and use as their first choice for interacting with data. While it shines in terms of support for flexible and fast data analysis, there are no established mechanisms in place to help the user easily and quickly reproduce the results from the visual analysis, not even the brushing results that the user has personally created. The lack of support for reproducibility is a general problem in IVA. It is not related to a particular view or brushing technique, although it is very likely that specific customizations depending on the technique will be needed to support reproducibility. Before we introduce our solution to the problem, we want to recapitulate the fundamental concept of linking&brushing and discuss why there are challenges with the reproducibility of results from the brushing

operation. This will help the reader better understand the need for the structured brushing space proposed in this work. One of the great things about interactive visual analysis is that users can select—with the help of interactive brushes—freely what they find interesting. We can employ a fundamental CMV dashboard, which encompasses a scatterplot, a curve view, and a rectangular brush on a scatterplot, to serve as an example to explain the problem with interactive brushing and discuss possible solutions. The dashboard is shown in Figure 3.1. Three dimensions of the previously explained meteorology data set [NOA14] are visualized. The scatterplot on the bottom-left is used to analyze two scalar attributes: the elevation on the vertical axis and the average temperature on the horizontal axis. The curve view at the top displays precipitation curves for all meteorological stations to support the observation of curves as a family of function graphs. Each station has one precipitation curve assigned, and precipitation values are measured monthly during one calendar year. Each curve in the curve view shows how the temperature value at the respective measuring station has changed over the measured period.

The small analysis task aims to get an insight into how a rise and fall in temperature affects the precipitation value in lower areas of California, i.e., with elevations ranging from 0 to 500 feet. Therefore, the user has initiated brushing in the scatterplot using a rectangular brush. A rectangular brush is usually created by anchoring the brush, i.e., placing the mouse at the desired position in the scatterplot, and then the rectangle of the brush is resized to the desired extent. In the shown case, the brush was first resized vertically to select elevation values between 0 and 500, and then horizontally to cover the temperature values ranging from 38 °F to 44.3 °F. Due to the high resolution of the visualization and the corresponding interface technology, the user had difficulties in quickly selecting the desired ranges on both axes. To study how temperature changes affect precipitation for the selected elevation range, the brush was swiped from left to right and back, keeping the same elevation while changing the temperature only, i.e., the user tried to move the brush along the desired path as precisely as possible to observe the changes in precipitation values visualized in the linked curve view. This way an interactive and iterative visual dialogue between the user and the system was established that quickly leads to relevant findings in the data, as documented by many examples so far. For the same reasons which mentioned above, difficulties also arise if users want to move the brush precisely or reproduce the brush's movement.

As shown in Figure 3.1 on the bottom-right, due to the hands' imprecise movement, the path of the actual brush is not as straight as the desired path, and as a consequence, some data items that have lower and higher elevation values than the desired ones were selected by the brush. The path from brushing would have been even more non-linear if the user had not taken special care to keep his hand steady while moving the brush. To repeat the movement of the brush along the same path is very likely a great challenge. Maybe we will not require precise placement of brushes if we do not mind that the position of the brush and a selected data subset deviates slightly from the desired ones. However, it is evident that there are challenges related to the brushing operation itself, mainly

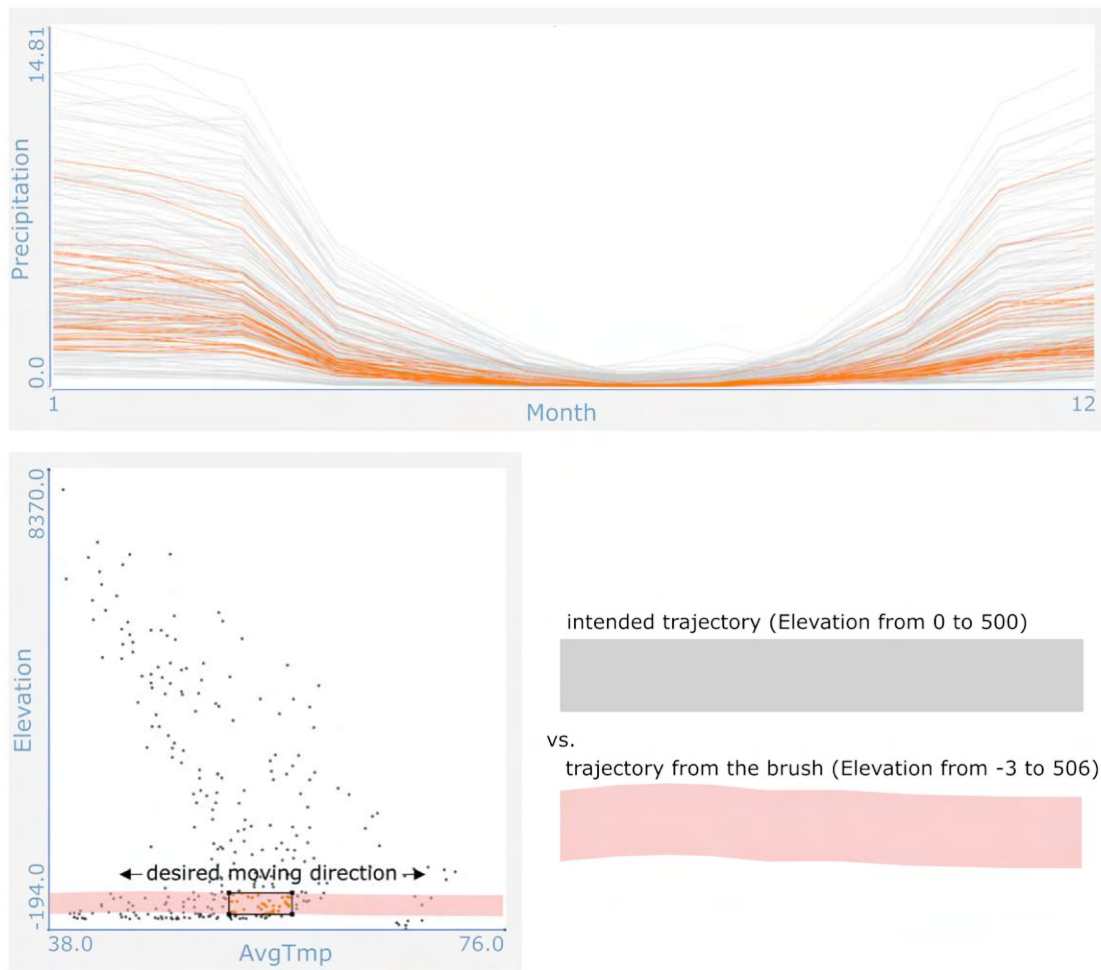


Figure 3.1: The rectangular brush created in the scatterplot (bottom-left) was moved horizontally to select only meteorological stations at a specific elevation level and with different temperature range. The data items currently selected by the moving brush are highlighted in orange in the scatterplot and in the linked curve view. We display the path taken by the brush on the scatterplot explicitly (shown in pink) and indicate the direction of the movement using arrows to communicate the brushing intention clearly to the viewer. In addition, on the right, we compare of the ideal straight path (gray) and the path taken by the brush (pink).

concerning the brushing precision and the reproducibility of the results from brushing. Considering the premises of interactive visual analysis (such as ensuring a fluid interaction between the user and the system), our first contribution to reproducible brushing and quantitative analytics is the structuring of the brushing space. The structured brushing space concept is adaptable and can be realized through various implementations. The core idea is to enhance the reproducibility of analysis outcomes by enabling users to exert

control over key brushing operations, including anchoring the brush, adjusting the extent of the brush, and controlling the movement of the brush.

	<i>Brush Anchoring</i>	<i>Brush Extent</i>	<i>Brush Movement</i>
<i>Unconstrained</i>	The user initiates the brush anywhere in the view, for example, on a scatterplot by specifying the top-left corner of a rectangular brush at an arbitrary position.	Any extent of the brush is possible and brush boundaries can be modified freely.	The brush can be moved freely.
<i>Constrained</i>	A “snap-to-grid” functionality is used to constrain the anchoring of brushes to grid vertices.	The size of the brush can be adapted in discrete, predefined steps only.	If moved, the brush assumes only grid-aligned positions.
<i>Automatic</i>	The user specifies a particular brush parameter, for example, a data-related property, and the brush is positioned automatically.	The brush resizes itself automatically due to certain constraints, for example, maintaining that a certain number of data items is selected.	The brush moves automatically, for example, following a user-defined animation procedure.

Table 3.1: Different aspects of the structured brushing space. From Radoš et al. [RSM<sup>+</sup>16], Figure 1.

The idea of the structured brushing space makes it possible to create brushes that can be reproduced accurately again, moved precisely in the view, or used to create exact selections. Instead of being burdened, for example, with placing the brush in a specific place or guiding it along a particular path, the user can partially leave this task to the mechanism that constrains brushing operations within the structured brushing space. The user decides on the properties of the structured brushing space that suits him and he would like to use. For example, he can proceed with the standard brushing, which we call unconstrained or unstructured, but when the need arises, he can easily switch to constrained or automatic brushing. The positive feedback that we got during the demonstration shows that analysts appreciate constrained brushing—it relieves them of worries about the brushing operation, and hence, it supports the analysis of the data displayed in the linked views. Technically, the structured brushing space is realized through mechanisms that influence the anchoring of the brush, its extent, and the movement of the brush. Table 3.1 describes examples of possible solutions for unconstrained, (partially) constrained, and (semi-)automatic brushing. In the following sections we show how some of the suggestions in the table can be implemented.



## 3.2 Snap-to Options for Brushing

The most common brush operations are anchoring, extending, and moving a brush. Easy reproducibility of these basic brushing operations can be enabled by exercising precise control over them. A common way to control objects in a view is through a snapping mechanism. Here, we propose the *snap-to* option for brushing. Once activated, the snap-to mechanism automatically acts on one or more aspects of the brushing operation, such as anchoring brushes only at predefined positions. We explain one concrete implementation in the following, the *snap-to-grid*.

### Snap-to-Grid Option

We connect the snap-to option to a grid. Using a grid, we structure the view space, or data space with the percentile grid, which we will talk about in the next section. The power and utility of grids are widely recognized, and they have become a standard feature in many applications. For example, invisible grids are often used in the background to organize and arrange tiles on a smartphone screen. Also, grids are handy whenever accuracy is essential. For example, artists use a grid to achieve proportional accuracy. See, for example, Figure 3.2 which demonstrates how the grid helps to create an accurate copy of an image.

Ward [War94] found that a grid can be used to measure the effectiveness of a visualization design. He also categorized visual elements into two groups: (i) those that can be directly employed to measure the effectiveness of the visualization, such as the count of displayed data items, and (ii) elements that are challenging to quantify, including keys, labels, and grids. These criteria necessitate subjective evaluation. He found that users interpret grids as reference points which help them to understand the data and their context. We share this opinion and add that grids can be effectively combined with different brushing techniques to realize a structured brushing space.

The analyst is often interested in finding relations to certain intervals along selected data dimensions, and a grid shown as an overlay in the visualization may help her in gaining additional insights into characteristics of the data, even before she initiates brushing. In this context, a grid can be referred to as meta-information, i.e., information which is not an explicit attribute of the data. We have adapted scatterplot and parallel coordinates to work with grids. Similarly, a grid can be added to various other visualization techniques with quantitative axes.

The grid presented here is freely configurable. The user can decide to show it on both data axes of a scatterplot or only for a single axis. If the grid is only activated for the vertical axis, only horizontal gridlines are displayed. If the grid is enabled only for the horizontal axis, then only vertical gridlines are displayed. If the grid is enabled for both axes of the scatterplot, as shown in Figures 3.3b and 3.3c, the user sees a two-dimensional grid composed of rectangular cells. A grid is presented to the analyst using a light gray gridline which is drawn beneath the semi-transparent data points. The user can alter the

### 3. REPRODUCIBLE BRUSHING

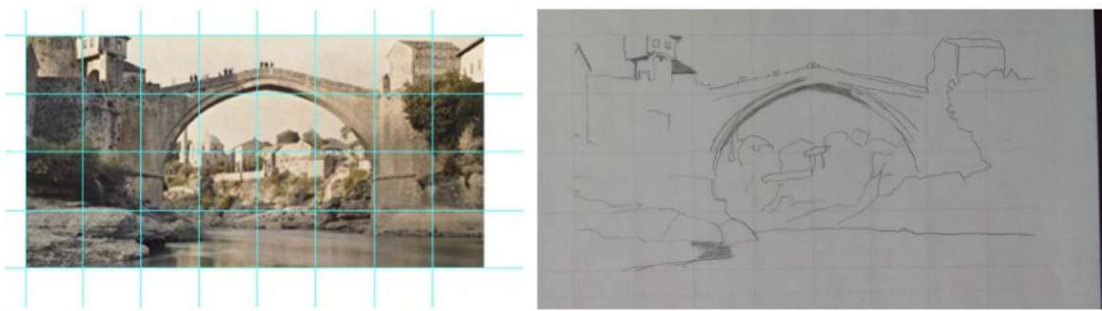


Figure 3.2: The grid is used as a visual aid to achieve an accurate line drawing from a reference image that shows the Old Bridge in Mostar, Bosnia and Herzegovina. One grid is drawn over the reference image, and another one—of equal ratio—is drawn on the paper. Grid cells help the painter to move the pencil more precisely so that all elements of the image retain their correct size and place in the repainted image.

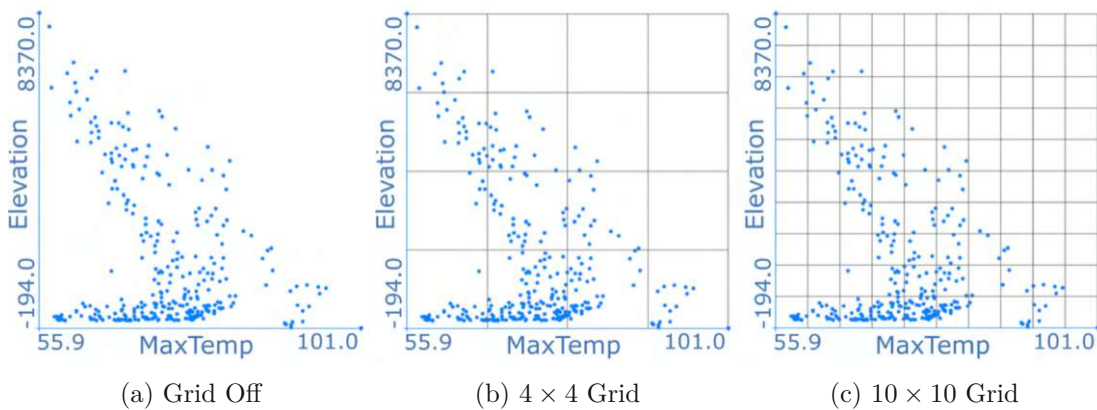


Figure 3.3: Starting from the (a) we show a scatterplot without a grid, a scatterplot with a  $4 \times 4$  grid, and a scatterplot with a  $10 \times 10$  grid. A grid shown as an overly supports understanding of the visualized data. Grids are rendered prominently only for better visibility in the printed image.

visibility of gridlines at any time. The gridlines are automatically calculated based on the desired number of divisions (strips) provided by the user. We propose using a  $10 \times 10$  grid as the default, but the user can adjust the number of divisions if the gridlines do not align with the data subspace(s) of interest. Also, it is possible to define a non-uniform rectilinear grid by placing gridlines at specific positions along the corresponding data axis by clicking with the mouse on the axis or by providing a list of numeric values for each gridline position through a dialog box.

Figure 3.3 shows three scatterplots used to visualize the same data dimensions. A quick look at the first scatterplot, which was intentionally left without a grid, instantly reveals that the two plotted data dimensions are not correlated, and that data items are not evenly spread. Approximately half of the data items have a low negative correlation,

being situated around the imaginary diagonal line that extends from the top left to the bottom right. Some data items have zero correlation, primarily those with low values in the vertical data dimension. Figure 3.3b shows a  $4 \times 4$  grid added as an overlay, which divides—visually—the view space formed by the two scatterplot axes into four equally spaced intervals, starting with the  $[0\%, 25\%]$  interval on the  $x$ -axis, and the  $[0\%, 25\%]$  interval on the  $y$ -axis. With the help of an overlaid grid, a more precise comparison of data distributions between different regions is enabled.

Now, with the grid creation mechanism in our hands, we can take the next step, which is to use the grid as an aid to control the brushes and confine brushing to reproducible shapes that can also be interpreted quantitatively. The *snap-to* option is well known, as provided in standard Office tools (such as PowerPoint and Excel), which can be set on user-added objects to align or snap them to the nearest intersection in the grid or other objects. We can also incorporate the *snap-to* function into interactive visual analysis by implementing a *snap-to-grid* for various brushing techniques. In this place, we use a grid to constrain all three core operations with brushing—anchoring, extending, and moving—or only a subset of them. For example, we can require that brushes are anchored at grid vertices, and we can confine the extent of brushes to match the size of the corresponding cell(s) of the grid.

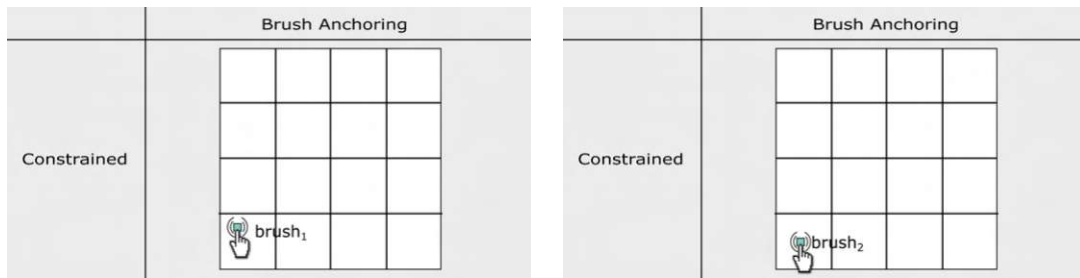
The concept of the snap-to-grid brush is given schematically in Figure 3.4, where three core operations with the brush in a scatterplot are presented: (a) brush anchoring, (b) brush-size adjustment, and (c) brush movement. Note that two different brushes ( $brush_1$  in the left column and  $brush_2$  in the right column) are used to clarify how the snap-to-grid mechanism works. Moreover, all three brushing operations are influenced by a structured brushing space using a regular  $4 \times 4$  grid and enabling the snap-to-grid option. Because the user understands how divisions produced by the grid relate quantitatively to the data axes, he also knows that if he creates a 2D/ rectangular brush in the bottom-left grid cell, he has selected the  $[0\%, 25\%]$  interval on the  $x$ -axis, and the  $[0\%, 25\%]$  interval on the  $y$ -axis, we could take advantage of any other grid division, the  $4 \times 4$  grid is just an example. Brushes snapped to the grid provide additional meaning to the user—they are quantitatively interpretable.

Through the anchoring mechanism (see illustration in Figure 3.4a), the brush is automatically bound to the center of the clicked cell. Nevertheless, this can also be set to be the closest vertex on the grid. Automation helps the user because it enables him to start brushing in exactly the same position, over and over again. Moreover, it can be more convenient for the user, as he does not have to be extremely careful with the positioning of the mouse because wherever he clicks, as long as he clicks inside the desired cell, the snap-to-grid mechanism will take care and properly anchor the brush.

After initiating the brush, the user commonly wants to adjust the extent of the newly created brush to select the area or data subset of interest. Suppose that the interesting data items are within the bounds of the bottom-left grid cell. Therefore, the user might think of resizing the brush to the size of the cell in question. With the snap-to-grid option enabled, the extent of the brush is constrained automatically, as displayed in Figure 3.4b.

### 3. REPRODUCIBLE BRUSHING

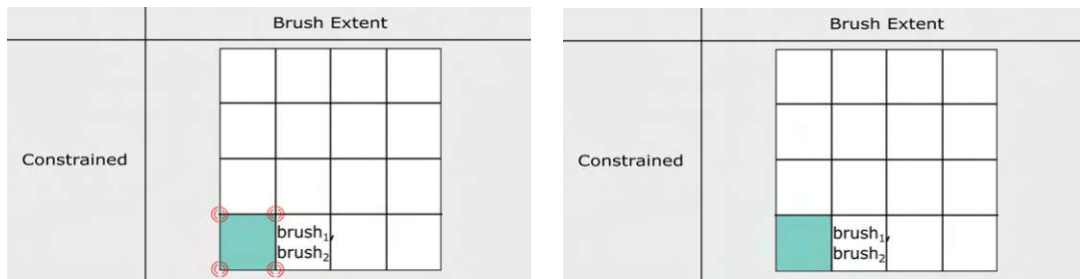
(a) Constrained Brush Anchoring



$brush_1$  is initiated at position  $(x_1, y_1)$ .

$Brush_2$  is initiated at position  $(x_2, y_2)$ .

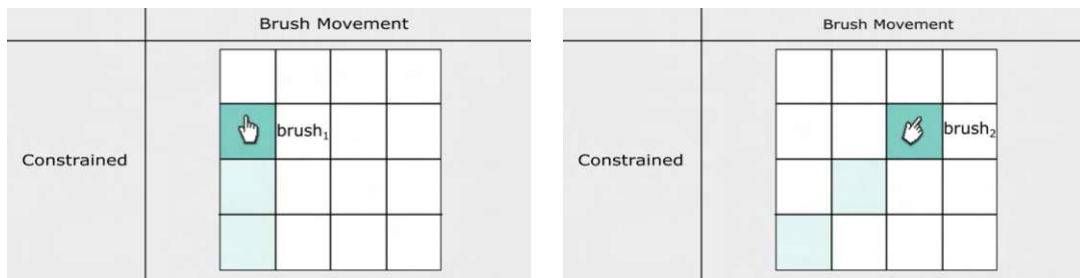
(b) Constrained Brush Extent



Brushes are automatically resized (red circles symbolize snapping to cell-vertices).

Brushes ( $brush_1, brush_2$ ) after creation. They have the same extent.

(c) Constrained Brush Movement



A sequence of steps:  $brush_1$  is moved vertically across two predefined intervals.

A sequence of steps:  $brush_2$  is moved diagonally across two predefined intervals.

Figure 3.4: The concept of constrained brushing using a rectangular grid and the snap-to-grid functionality. Two brushes are shown for better clarification. **(a)** Note the different click-position for  $brush_1$ , and  $brush_2$ . However, both brushes are automatically anchored to the same cell. **(b)** The extent of the brushes is confined to the extent of the cell in which the brushes are anchored. **(c)** The movement of the brushes is allowed only in a specific direction (for example, through the cells of the grid); the current brush position and the two previous positions (brightly colored) are shown.

Constraining the brush’s extent does not mean permanently fixing the brush’s size. We provide the flexibility to modify its extent further. For example, the user can stretch it horizontally so that the new width corresponds to the width of two grid cells or any other size, a multiple of a cell width.

The constrained movement is displayed in Figure 3.4c. The movement of *brush*<sub>1</sub> is constrained vertically, while *Brush*<sub>2</sub> takes only predefined intervals in the diagonal direction. This can be relaxed to cell-by-cell movement in any direction. Even an unprecise interaction in the brushed view will result in the expected brush movement. This allows the user to concentrate on the linked views, knowing exactly which intervals are selected, without the need to paying attention to value-accurate brushing.

It is important to note that we have retained the existing brush-handling functionality implemented in ComVis [MFGH08], such as moving the brush, by holding it with the pressed mouse button, and adjusting its size (by selecting and moving one of the rectangle’s edges or vertices). Finally, very importantly, the snap-to-grid option can be disabled or enabled at any time for the selected brush, and the user can continue using the brush without any restrictions.

To explain the snap-to-grid mechanism schematically in Figure 3.4, we used a Cartesian grid. However, the user can create or choose between different regular or rectilinear grids, we did not experiment with other more grid types such as structured grids or unstructured grids. In Section 3.3 we introduce the *percentile grid*, a new rectilinear grid type that can automatically position its gridlines according to the statistical characteristics of the underlying data.

### Snap-to-Brush Option

In addition to the snap-to-grid option that can be used with many brushing techniques combined with an overlaid grid, the snap-to-brush option enables additional brushing opportunities in specific cases. Depending on the geometrical shape used for the particular brush, the snap-to-brush can be implemented differently. It is good to think of these options as connection points that one shape provides (e.g., a rectangle) so that other shapes (e.g., a circle) can be connected to it. For instance, in the case of a rectangular brush, we considered snapping to one of its vertices, edge centers, or the center point. For the circular brush, we implemented only a snap-to-brush for the center point. Other options are also possible depending on the user’s needs. The snap-to-brush option can also be activated for a short time as an aide if we need to position two brushes relative to each other. In this context, it has proven very useful to support the creation of reproducible composite brush constructs, where two or more brushes are combined with logical operators to select specific subsets of data items.

Another valuable option for brushes is to enable editing of the anchor (e.g., the center point) and the extent (e.g., the radius) of the brush for precise positioning and selection of data items. For instance, to enable the reproducible anchoring of the circular brush in a scatterplot, the user specifies two components of a center point by entering two float

values, each within the range of values according to the data dimension mapped on the corresponding scatterplot axis. In addition to anchoring the brush to a specific position in the data space, we offer the option of entering the coordinates of the point  $(x, y)$  for the center point, one or the other option can be used depending on the situation. This option manually specifies the brush shape by entering the required values. Although we have only implemented it for the circular brush, it can easily be added to the rectangular brush in a scatterplot and other more complicated shapes, which can be described mathematically. In addition to helping users precisely set the brushes, all these options also enable the reproducibility of brushes. However, IVA users traditionally prefer to work with brushes on the go, i.e., directly interacting with the shape (geometry) of the brush in the view, most often using a mouse pointer, because it speeds up the analysis and, in most cases, although less accurate, the qualitative inference is good enough.

### 3.3 Percentile Grid

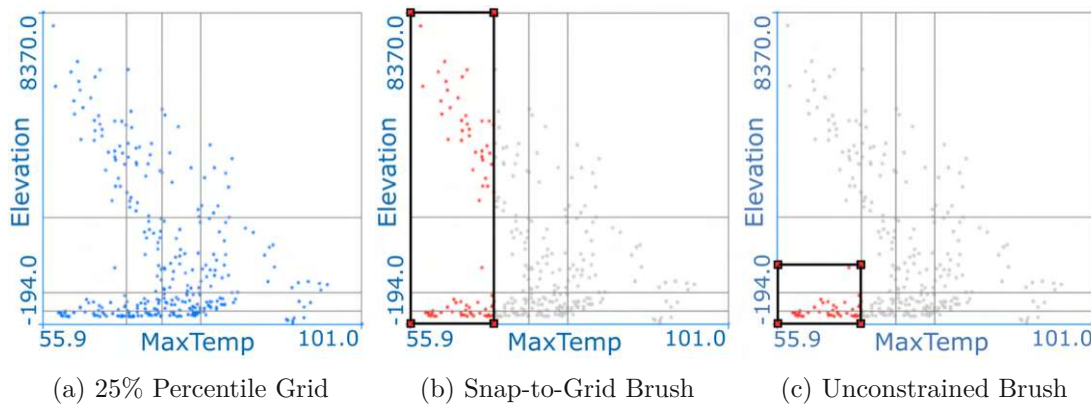
With the help of descriptive statistics, it is usual in (computational) data analysis to either do a value-based analysis, or a rank-based analysis. The latter could, for example, be enabled through quantile filters [Che03] or statistical estimators [KFH10]. Hence, we suggest to also provide brushing opportunities which match these analytics approaches.

Using a regular grid corresponds to a value-oriented perspective. Often a rank-based perspective is also very useful. An example would be to compute the Spearman correlation [Spe87]. Instead of selecting all items that correspond to a certain range of values, we are interested in a certain number of data items, for example, the top 10% of all data items. If we define the grid so that each division on an axis separates a certain percentage of the items, we create a *percentile grid*.

The percentile number defines the number of automatically created subdivisions, i.e., the number of strips along a selected axis in a scatterplot. The 25% percentile is used by default. In statistics, it is common to divide a data set into quartiles. The user can change this number. Default percentile numbers from which the user can choose are 1%, 2%, 4%, 5%, 10%, 20%, 25%, and 50%—as these numbers allow for creating an even number of divisions in a grid. If a 25% percentile grid is enabled, we first calculate the 25th percentile, then the 50th percentile, and finally the 75th percentile, i.e., four divisions are created. The user can override the automatic grid-division process by providing a list of percentiles to be used to form the grid. For example, entering [15, 20, 20] would result in four subdivisions along the corresponding axis, where the first one contains 15 percent of the lowest values, the second and the third one are for the 35th and 55th percentiles respectively, while the last one contains 45 percent of all items with the highest values.

Each vertical and each horizontal strip of the scatterplot in Figure 3.5a, for example, contains exactly 25% of the data—note the different sizes of the grid cells. Snap-to-grid brushing using the percentile grid has a different meaning compared to value-based brushing using the Cartesian grid. Brushing all left-most cells, snapped to a 25% percentile grid, we know, again quantitatively that we have selected the 25% lowest values with





Data View					
Show Selection					
	Name	Latitude	Longitude	Elevation	Maximum Te
1	WILLITS 1 NE	39.41	-123.1	1350	66.0
2	TWITCHELL DAM	34.98	-120.15	582	67.4
3	PACIFICA 4 SSE	37.6	-122.13	475	62.0
4	BEN LOMOND NO 4	37.8	-122.8	420	67.7
5	GRIZZLY CREEK STATE PARK	40.48	-123.8	410	63.0

(d) A view showing data items brushed in Figure 3.5c.

Figure 3.5: (a-c) Starting from the left side, we show a scatterplot with a 25% percentile grid, a scatterplot with a brush snapped to a percentile grid and a scatterplot with a user-defined (unconstrained) brush. d Data View is opened for further inspection of the data subset selected by the brush created in Figure 3.5c (attributes are sorted descending by Elevation).

respect to the dimension that is mapped to the horizontal axis (see Figure 3.5b). Moving the brush along the grid from left to right, then, would accordingly select consecutive portions 25% of all items.

The analyst may benefit from the grid even if the constrained brushing is not enabled. The percentile grid reveals some insight into the data distribution. Additionally, an overlaid grid can also assist the navigation of the brush over the visualized data items or help the user select data items more precisely. The brush created in Figure 3.5c is unconstrained, i.e., it is a regular brush free-hand created by the user. The overlaid percentile grid was of great help to the user in making the quantitatively meaningful selection of data items. The brush selects quite accurately all data items that belong to the 25th percentile with respect to the horizontal data dimension and the 50th percentile with respect to the vertical data dimension; additionally, the user decided to adjust the height of the brush by dragging the top edge of the brush rectangle upwards with a mouse in order to include additional data items in its selection (with an elevation value below 1500). If users make precise selections using brushes, they often display the selected data in an additional view which shows original numbers to confirm that they



have selected the preferred data (in this case, the Data View was opened, which provides a tabular view for the selected data, and it confirms that all selected data items have an elevation lower than 1500). As far as understanding the brushes, basically what data we brushed, constrained brushes are much more informative than the classic ones, which are unconstrained (however, to confirm this, an additional user study would be needed).

### 3.4 Grid Extensions for Parallel Coordinates

Up to now, we have explained the regular grid and the percentile grid for a scatterplot. In the following, we present a grid extension as a general extension to the well-known visualization technique of parallel coordinates.

In parallel coordinates, the coordinate axes are arranged parallel, side-by-side, and the user can decide which dimensions of a multi-dimensional data set are mapped to the enabled axes. Moreover, we also provide the flexibility to choose whether grids are enabled for all parallel axes, a selected set of them, or a single axis. Like in a scatterplot, it is possible to use a different grid on each axis (divisions for each grid are calculated using data from the underlying data dimension). The idea is the same: to create a grid, we divide the axis into divisions whose width depends on the specification of the grid. The user can also provide a list of percentiles to specify the grid. If the user enables the percentile grid without specifying a percentile for the division, 25% percentiles are used by default. The percentile number defines the number of automatically created sub-divisions, i.e., the number of strips in a scatterplot or number of bins in parallel coordinates—in a scatterplot, we draw horizontal or vertical strips, depending on whether the selected dimension is mapped to the  $x$  axis or to the  $y$  axis, and in parallel coordinates plot, we divide an axis accordingly in bins.

Enabling the 25% percentile grid in parallel coordinates will result in displaying four different bins. An example is given in Figure 3.6; The percentile grid is enabled only for three coordinates axes: Elevation, AvgTemp, and MinTemp. By default, the grid is rendered semi-transparently (see grids enabled for AvgTemp and MinTemp in Figure 3.6). The grid for the Elevation axis is rendered prominently because the mouse cursor is placed over it to show details about the used grid. On-demand, the numbers describing the used percentiles for each bin are displayed. For other grid types, the values shown refer to the number of data items covered by each bin. We implemented the details-on-demand option only for grids in parallel coordinates plot, but it can also be added to other plots that implement the grid techniques proposed here. We show data dimension Elevation both in parallel coordinates and in the scatterplot (see Figure 3.5) to help the reader in comparing how the same 25% percentile grid is visualized in the two different charts.

Similar to histograms over parallel coordinates used by Hauser et al. [HLD02]), our percentile grid also reveals where data items accumulate along the coordinate axes. A problem with the percentile grid is that the higher percentage of data items that fall into a certain percentile, the smaller will be the size of a bin corresponding to that percentile (for an example, see the 25th percentile and the 50th percentile for axis Elevation in

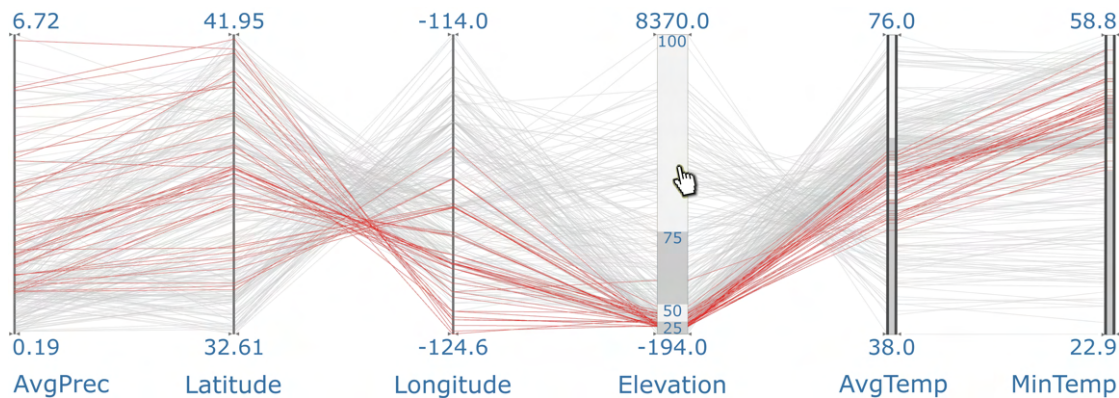


Figure 3.6: Extended parallel coordinates: The percentile grid is laid over Elevation, AvgTemp, and MinTemp (each division contains 25% of data values). The mouse was placed over Elevation to display percentile values. The brushed lines are highlighted in red, however, the brushing was done in the linked scatterplot (see Figure 3.5c).

Figure 3.6). This problem affects the mentioned histogram solution less because all histogram bins have an equal extension.

In the case of a regular (Cartesian) grid that we propose for supporting the value-based analysis, we also generate multiple bins of equal size. In this case, the visibility depends on a user-defined number of sub-divisions, and it can be improved by reducing the number of bins. The problem of exploring small-size bins remains, and we need a more convenient method to support visual exploration. We found *dimension zooming* proposed by Fue et al. [FWR99] to be one simple but also valuable interaction technique that can help the user in interpreting data items within a narrow bin. They use this distortion technique in parallel coordinates to display the brushed data in full view. The whole display space is used to visualize the brushed subset. To keep track of the context, they display a mini-map showing the position of the zoomed subset in relation to the complete data set. The zooming operation is done by scaling up independently each of the visualized data dimensions concerning the extent of the brushed data values. We implemented the dimension-zooming option for grids in parallel coordinates, which helps, for example, if the user is interested in the data of a specific quartile only. Three ways for defining the scale factor for zooming to grids are available: (i) scaling regarding the data range defined by the brushed data items within the selected bin, (ii) scaling regarding the data range defined by all data items within the selected bin, and (iii) scaling regarding the data range that the selected bin covers on the axis. The difference between options (i) and (ii) is whether we are interested in observing only the brushed data within the selected bin or we want to see all data items—the first option usually leads to a higher scaling factor, i.e., the narrower the data range the higher the scaling, as shown in Figure 3.7. In Figure 3.7b the data range that is displayed on the coordinate axis is wider than the range displayed in Figure 3.7a, because the latter shows only the brushed data. The user is free to select one or more bins on a single coordinate axis, and the scaling value

### 3. REPRODUCIBLE BRUSHING

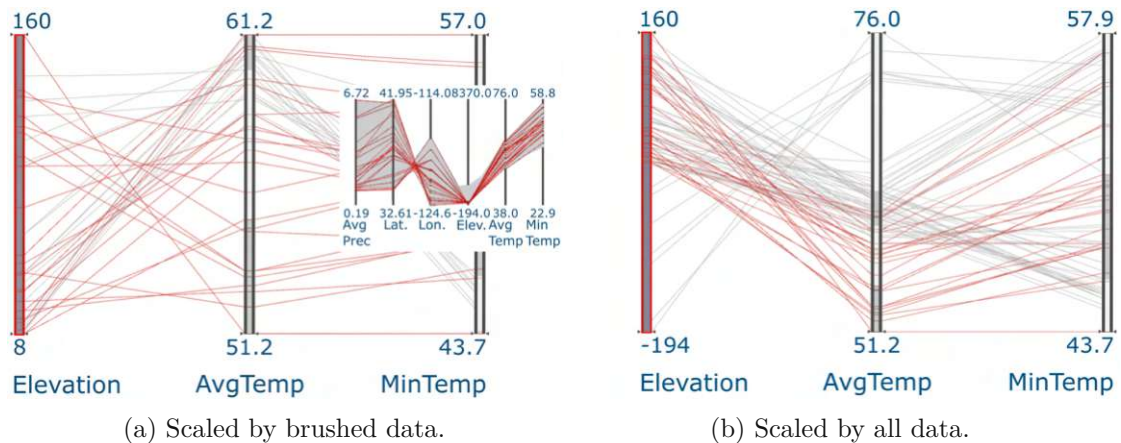


Figure 3.7: The first 25% percentile bin on the Elevation data axis (highlighted in red) is maximized for the analysis (see Figure 3.6 for comparison, which shows the Elevation, AvgTemp, and MinTemp data axes prior to scaling). Scaling of all other parallel coordinate axes is done concerning the scaling of the selected bin. In **(a)**, only the brushed data contained within the analyzed bin are visible (option (i)), while **(b)** shows all data within the selected bin (option (ii)). The range of data displayed is shown next to each axis. Additionally, a small mini-map provides an overview of the range of the zoomed subset of brushed data (red lines) relative to the range of all brushed data (grayish area).

will be adjusted considering all the selected bins (if the selection includes bins that are not adjacent, the referenced subspace used for scaling will be automatically adjusted to display all data from the analyzed bins).

The snap-to-grid brush option is also implemented for parallel coordinates. Brushing in parallel coordinates is commonly done by marking a particular subset of data items within a single data dimension. Since the proposed grids only work for a single data axis in parallel coordinates, the brush snapped to the grid only considers the dimension of the data for which the respective grid is enabled (the brush can be anchored at the center of the bin or bin edges). With the help of the percentile grid, the analyst can create statistically meaningful brushes. For example, in order to select 10% of all data items with the highest elevation value the extent of the brush created in Figure 3.8 is constrained to the edges of the top most bin created for the Elevation data axis. This example also demonstrates the clear need for an option that allows the user to zoom the grid, or some other type of axis scaling, for the reasons explained above. The lowest deciles created for the Elevation axis can hardly be separated visually.

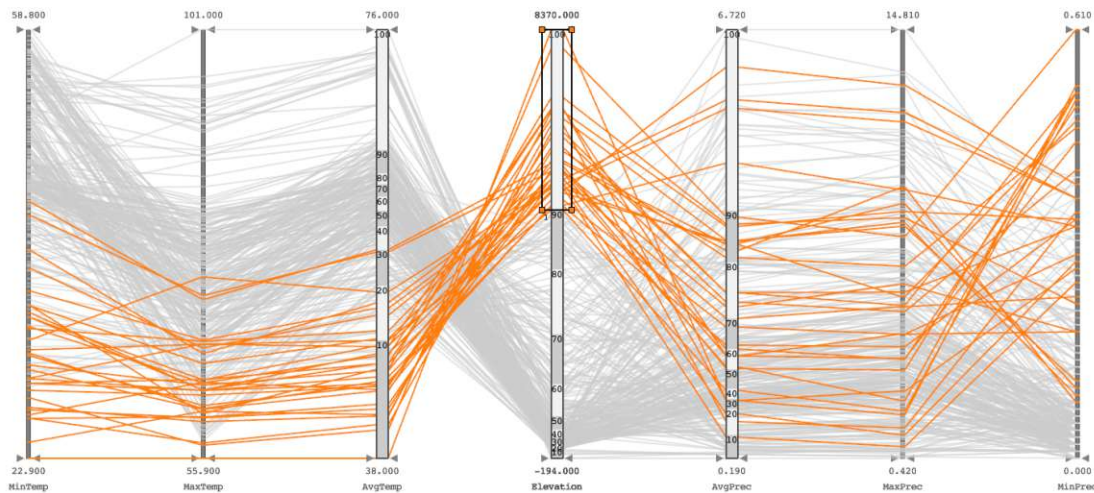


Figure 3.8: Snap-to-grid brushing in parallel coordinates. The 10% percentile grid is enabled for three data axes: AvgTemp, Elevation, and AvgPrec (each of these axes is divided into 10 bins, where each bin contains the 10% of all data items). The user can easily observe correlations between visualized deciles using the snap-to-grid option. The topmost bin created for the Elevation data dimension is brushed. The user can move the brush, and in each step, the brush will select the adjacent decile.

### 3.5 Percentile Brushes

The percentile grid discussed in the previous section proved to be very useful. It provides quantitative insight into the distribution of the displayed data and, with snap-to-grid brushing options (which works for any grid), is a helpful tool for enabling rank-based interactive visual analysis. The snap-to-grid functionality facilitates data exploration by allowing users to select meaningful subsets of data items precisely and quickly, like the first 25th percentile for the selected data dimension. Using the snap-to-grid brushing option, users are conditioned to keep the brush movement tied to the grid, which means that they cannot position the brush anywhere on the data axis and then select, for example, 25% of data items around the current brush position. A solution to such user requests is supported with the *percentile brush*, an advanced brush, which is based on the structured/informed brushing space. The percentile brush considers the underlying data in a similar way as the *percentile grid* does, but it does not have to be snapped to the grid for brushing data quantitatively. Some constraining is still necessary. The user can position the anchoring point of the percentile brush freely, but its extent is always constrained. By automatically constraining the extent, this brush always selects a predetermined percentage of items, such as 10%. As the brush is moved, its extent is continuously adjusted to maintain the selected percentage of items, as explained in the next section. Users can always interpret their percentile brushes quantitatively, as they represent, i.e., select, a fixed percentage of the data. We implemented two standard shapes for realizing percentile brushes in a scatterplot: the rectangular and circular



percentile brushes (we also experimented with a square brush shape, the implementation of which is very similar to the circular brush shape, but we did not use the square shape in our demonstration). The percentile brushes are a new technique for brushing scatterplots. The circular percentile brush and the square percentile brush work in two dimensions, while the rectangular percentile brush considers a single data dimension.

#### The Rectangular Percentile Brush

The two-dimensional scatterplot has one horizontal and one vertical data axis. When creating the rectangular percentile brush, the user can decide whether the brush considers the data distribution in the horizontal or the vertical dimension, and the brush will be anchored on that specific axis. The scatterplot in Figure 3.9 has two rectangular percentile brushes created for the horizontal dimension, and the scatterplot (d) has two rectangular percentile brushes for the vertical dimension. Each of these brushes selects 25% percent of all data items. Because they are positioned at different positions on the respective axis and because, in this case, the data distribution near their anchor points is different, all percentile brushes shown have different (width) extent. For comparison with the snap-to-grid technique, scatterplots at (a) and (c) of Figure 3.9 show the 25% percentile grid, and the traditional rectangular brushes are snapped to grid vertices to select an equal number of data items as the percentile brushes in the scatterplots in (b) and (d)—mirrored brushes intentionally select the same data items. We show the rectangular percentile brush as a solid yet semi-transparent area to visually emphasize the brushed area and distinguish this brush from the standard rectangular brush. Snapping brushes to grid elements is a powerful feature, and we want to have it for the percentile brush as well. For precise positioning, the center of the percentile brush can be anchored at the center of a cell or the cell vertices, and the percentile brush will continue to update its extent automatically whenever it moves to another position.

Once they are created, the rectangular percentile brushes can be (freely) moved only in one direction, along the horizontal or vertical axis, i.e., along the axis on which they are anchored. The extent of the brush rectangle changes only in one direction. Our implementation supports users' cognition of changes while the rectangular percentile brush is moved. For the brush created on the horizontal axis, as shown in the scatterplot in Figure 3.9, the width of the brush rectangle is updated depending on the data distribution under the brush. The width of the brush-rectangle increases if the brush is moved to a low-density region, while the width decreases if the brush is moved to a high-density region. The extent of a percentile brush communicates an important message about the distribution density of data items under the brush. This is clearly observable in the scatterplot. Although both brushes shown have selected the same number of data items, their extents, i.e., the selected ranges on the axis are very different. Knowing how the rectangular percentile brush works, the user can immediately distinguish, for example, whether the rectangular percentile brush selects horizontal or vertical data dimensions in a scatterplot when looking at the view or screenshot in a report. One exception is the percentile brush which selects 100% of all data items because, in such a case, assuming

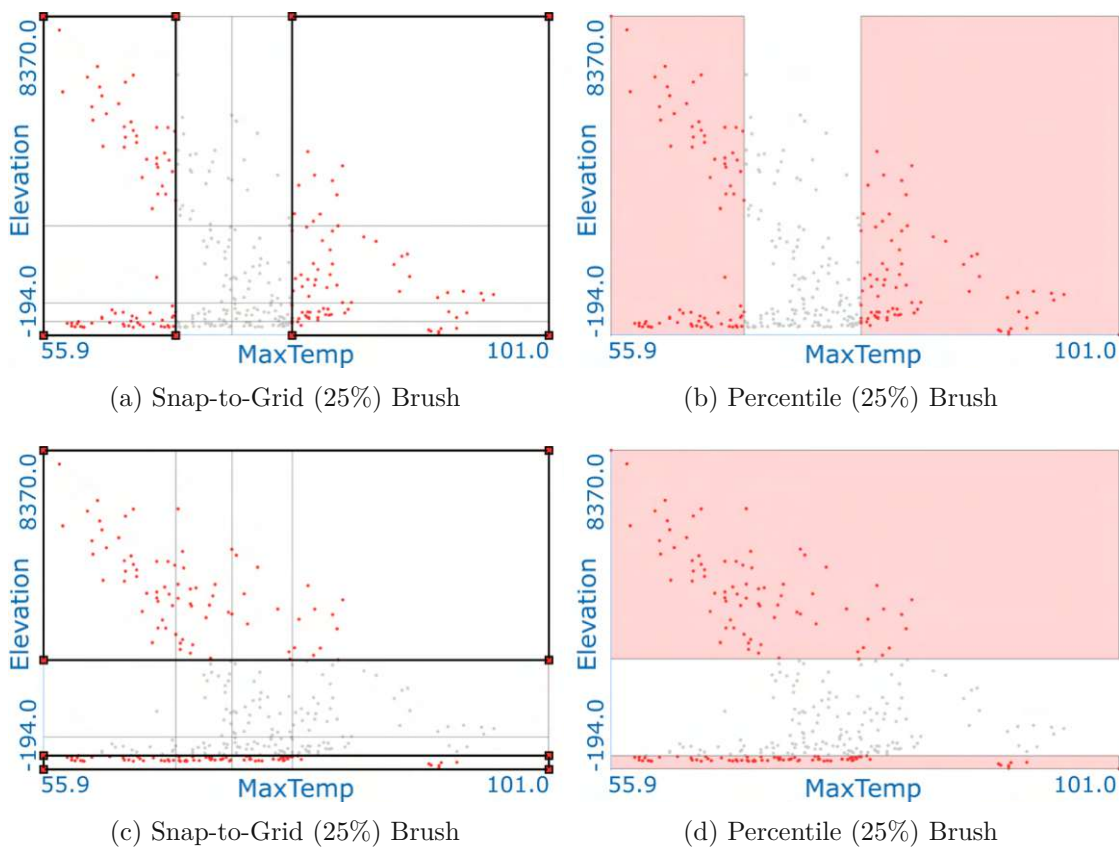


Figure 3.9: **(a)(c)** Quartile brushing in a scatterplot enabled with the 25% percentile grid and the snap-to-grid option for brushing. **(b)(d)** Quartile brushing in a scatterplot using the 25% rectangular percentile brush. Scatterplots (a) and (b) show brushes created in the horizontal dimensions, while scatterplots (c) and (d) show brushes created in the vertical dimension.

that the aspect ratio of scatterplot axes is equal, the width and height of the brush will be the same.

The user usually interacts with traditional brushes directly in the view, for example, he chooses an arbitrary point as the top-left corner of the rectangular brush and then extends the brush rectangle to the desired size, or he drags one edge of the brush rectangle to change its width to include additional items in the brushed data subset. The corners of the created brush-rectangle define the range of the selection, i.e., all data items within the corners of a common rectangular brush are selected. The percentile brushes are a little different to handle due to their semi-constrained nature. The user can move the percentile brushes freely by selecting them and holding them with the mouse, as shown in Figure 3.10, but there is no such action as pulling the edge of the brush to change its size.

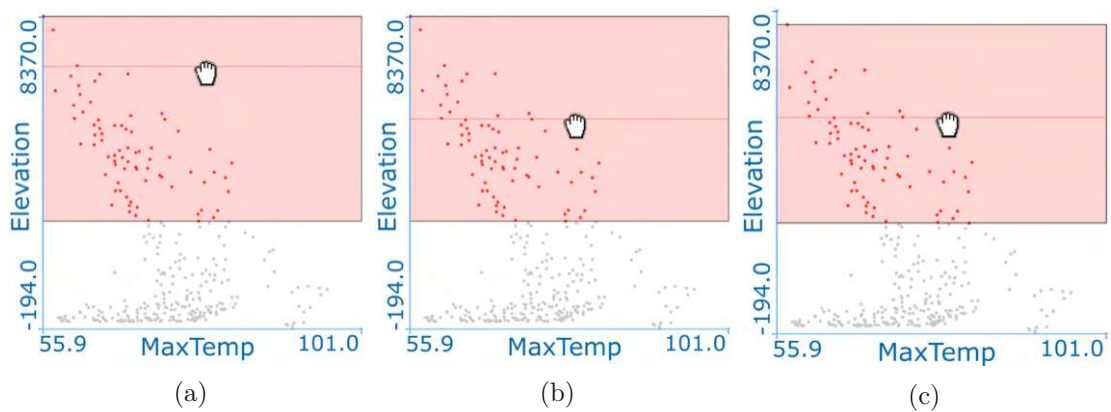


Figure 3.10: **(a)(b)(c)** Brushing the vertical axis in a scatterplot using the rectangular percentile brush. The brush that selects 25% of all data items considering Elevation data dimension is shown at three different positions. Optionally, the seeding line can be displayed (see horizontal line under the mouse) to help the user mentally relate the current position of his hand in the view with the exact position on the coordinate axis.

Percentile brushes consider data items around the current mouse position and one off-screen parameter for the brush size. To create a new percentile brush, the user sets the percentage value  $p$  of one off-screen parameter and chooses an arbitrary point (such as near the middle of the data items to be selected) by pressing the left mouse button. The percentage value is set by default to 10%, and the user can subsequently change this value as needed (for convenience, via the provided user interface, or by moving the mouse wheel), and the brush immediately adapts its extent. Creating the percentile brushes (assuming the user agrees to use the default value) is as easy as a mouse click. The percentile brush starts adjusting its extent automatically, but it only does so after checking its off-screen parameter's percentage value  $p$ . The value  $p$  is internally converted to an integer number  $m$ , which defines the exact number of data items in a local data subset  $D_s$  that the brush should select.

The calculation of the brush starts by obtaining the coordinate position of the click-point  $\mathbf{s} = (s_x, s_y)^T$ . For a brush on the horizontal axis, then the brush needs  $s_x$  value, and for a brush on the vertical axis value is needed. The viewpoint, i.e., mouse position, is then converted to the data value  $d_i$  (regarding the range of data values in data dimension  $D$  that is mapped onto the coordinate axis on which the brush is created). Finally, in the sorted list of values (internal representation of data dimension  $D$ ), we find the index  $i$  of the actual data value closest to  $d_i$ . Now we can quickly obtain the remaining  $m - 1$  data items for creating the data subset  $D_s$  (we do this by examining  $m - 1$  neighbors of  $i$ ; for example, the next data items will be either the data items on the index  $i + 1$  or on the index  $i - 1$ , depending on which of these two data values is closer to the data value on the index  $i$ ). The calculation of the percentile brushes can be done reasonably fast in ComVis [MFGH08], thanks to the clever implementation of its data warehouse. The two most important details concerning data management in ComVis are: (i) the



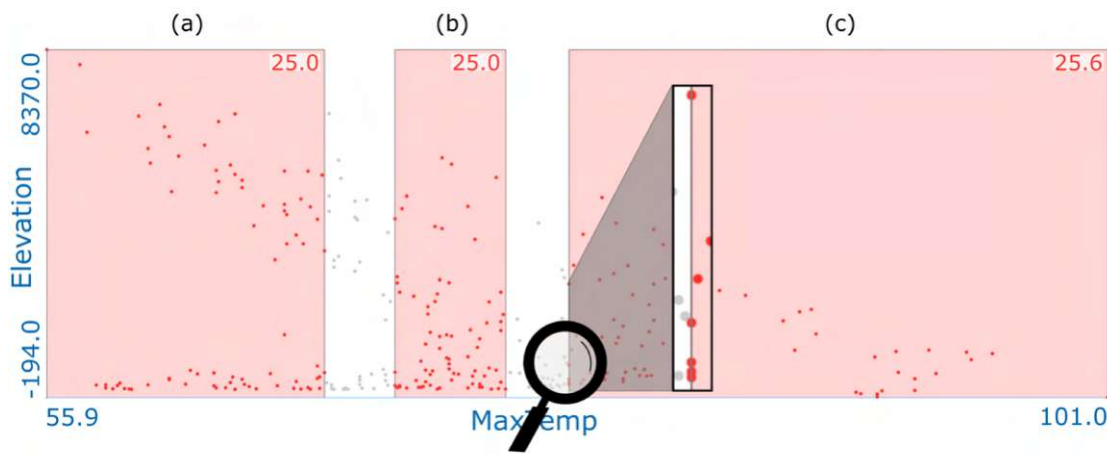


Figure 3.11: There are three rectangular percentile brushes in the scatterplot (the view is stretched horizontally to get a better overview). Each brush is set to consider 25% of all data items. The brush (c) has selected more items, exactly 25.6%, because there are several data items (sharing the same MaxTemp value) on its left edge (a magnifying glass highlights this area).

data are saved in columns (all data items in a column are of the same data type while different columns can contain different data types), and (ii) to ensure efficient data query ComVis sorts all columns at the beginning (in order to keep the original row information it creates an additional indexed column for each column).

There is a limitation on the accuracy of the rectangular percentile brush. After  $m$  closest data items to the mouse position are found, the brush algorithm stops. The lowest  $r_{min}$  value and the highest  $r_{max}$  value from the selected data subset  $D_s$  define a closed interval  $[r_{min}, r_{max}]$  this brush will represent on the axis. The shape of the brush is a rectangle, and the brush will select all the data items within the rectangle ( $r_{min}$ , and  $r_{max}$  are used to define the vertices of the brush rectangle). We have a typical classification problem with two classes and 1-dimensional feature space. Moreover, the brush uses 1D decision boundaries (lines). Therefore a decision boundary, i.e., the division between the brushed region and the context, can be ambiguous—it is not always a clear “single data item” cut. Figure 3.11 illustrates the problem. Brush (a) and brush (b) select exactly 25% of all data items. Brush (c) has a problem because there are several data items with the same value on its left edge, and so the brush has to decide whether to include them all or create a selection with fewer than  $m$  data items (we decided to include  $\geq m$  values). Numeric annotations (see the numbers in the upper right corner for each brush) can be optionally displayed for the percentile brushes, which inform the user about the actual percentage value of the selected data for each brush in the case of an ambiguous decision boundary. In the worst case, if all data values of the brushed dimension are equal—displayed as collinear points in the scatterplot—the rectangular percentile brush will always select 100% of the data, regardless of the percentile parameter

set. If  $r_{min} = r_{max}$  the brush-rectangle will appear as a line in the scatterplot. The user will probably recognize such extreme cases by looking at the distribution of data items in the scatterplot before he initiates brushing. He will use the percentile brush only to confirm his observation.

Conveying the movement with static images is no easy task, so to explain how the rectangular percentile brush moves, we decided to use three selected images taken from a continuous brush movement. Figure 3.10 shows the same brush positioned at three different positions in a scatterplot. Notice the significant change in the user-controlled mouse pointer's position between (a) and (b), but the brush did not move. The explanation for this is simple: the mouse's position serves as a starting point for the calculation of the brush extent and does not necessarily have to be aligned with the center of the brush, which depends on the underlying data distribution. The computed subset  $D_s$  is not updated if the mouse is moved from (a) to (b). In contrast, although the difference in mouse position between (b) and (c) is minimal, the subset  $D_s$  is updated—one point has been excluded from the selection while a new point has been added to the selection (both values  $r_{min}$  and  $r_{max}$  have changed). The movement of the rectangular percentile brush, in most cases, will not be smooth, but it is more of a jerky movement. This makes sense for the user who wants to examine the distribution and scattering of the displayed data or perform a rank-based analysis.

#### The Circular Percentile Brush

The circular shape of the brush is especially suitable for scatterplots because it can take advantage of the Euclidean coordinate system and two-dimensional plane defined by the orthogonal axes of the scatterplot. As with other traditional brushing techniques, the primary task of the circular brush is to allow users to select one or more data items shown in a scatterplot. Interaction with a circular brush is usually implemented so that the user selects an arbitrary point in the scatterplot as the center of the circular brush and then expands the radius of the brush to include more data items. The circular brush works well in a scatterplot if the user wants to select an area within a certain perimeter relative to the mouse cursor's position or if the user wants to select a round-shaped cluster of data items. The traditional circular brush does not support the user to specify how many elements should be in the selected subset, whether it will be just one, more, or a certain percentage of data. The selection made by this brush is not defined by the number of data items but by the area of the circle. All data items within the area will be selected regardless of their number. The user has to assess how much data is inside the brush qualitatively, but in most cases, it is not easy to estimate that, especially if, for example, there are many data items or if the user has to think about it while quickly moving the circular brush in the scatterplot. The traditional circular brush is not suitable for a task where the selected subset should have the same number of items at all times (also when a brush is moving), which means that including a new item replaces the old one. The rectangular percentile brush we presented in the previous section supports such a task by automatically adjusting its extent to keep the same number of elements in the

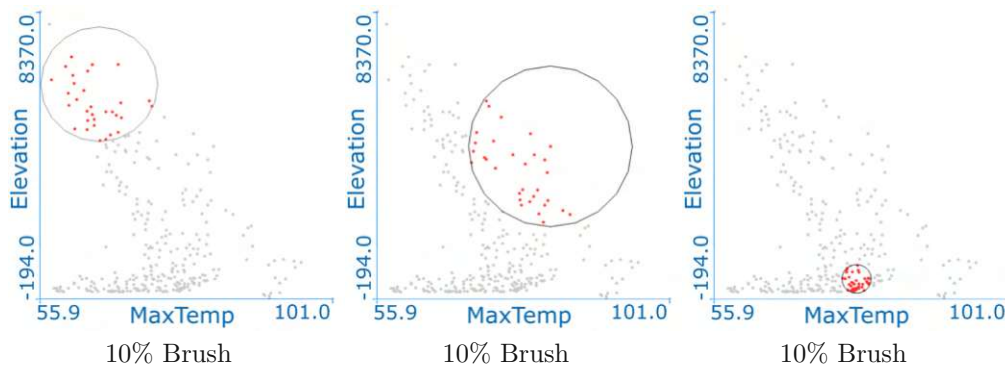


Figure 3.12: Brushing with the circular percentile brush in a scatterplot. Changing position using fix percentage value. The user wants his brush to select 10% of all data items all the time. She can rely on the brush mechanism because the percentage value is set to 10% (details in text). Three different figures display the same brush moved to three different positions. Notice how the circumference of the brush is automatically adjusted depending on the data distribution below the brush.

selection as it moves. Adjusting its extent, the rectangular percentile brush considers only one data dimension, and also, its shape is not well suited for selecting circular clusters of data. We now present another advanced brush, which shares a similar design choice with a rectangular percentile brush, but it has a circular shape. This new brush considers data items from both data axes of the scatterplot to maintain its circular shape, and because it selects a certain amount of data, expressed as a percentage of the total data that the user wants to select, it also supports rank-based analysis. Figure 3.12 shows a circular percentile brush in a scatterplot. A rank-based analysis was enabled using the circular percentile brush, which selects 10% of all data items in a scatterplot closest to the mouse position, i.e., to the center of the brush.

Before going into details about the circular percentile brush, we want to briefly explain why the circular brush shape should be used carefully in a scatterplot, especially when the scaling of each individual data dimension is different (that is, if the axes do not show the same range of values), or when the axes of a scatterplot display different quantities (as in Figure 3.12). Such cases often arise if analyzing multidimensional datasets consisting of mixed data dimensions. If the units and/or scaling of the data dimensions are not the same, one should be careful with estimating the distance based on the circle's radius. We are used to thinking about the distance between objects on the 2D plane in terms of Euclidean distances. Recall that the Euclidean distance between any two points on the real line can be measured with a ruler or by calculating the absolute value of the numerical difference of their coordinates:  $|x_1 - x_2|$ . For two-dimensional points, the distance between points is often calculated with the help of the Pythagorean theorem, whereas for points in higher dimensions, the distance is calculated using the Euclidean distance formula. For two-dimensional points given by Cartesian coordinates, the following formula can be used to calculate the distance between point  $p = (p_x, p_y)$

and the center of the selected data subset  $c = (c_x, c_y)$ :

$$ED_{p,c} = \sqrt{(p_x - c_x)^2 + (p_y - c_y)^2} \quad (3.1)$$

Given the center point and circle radius, we can utilize this formula to calculate whether a point lies inside, outside, or on the circle. This approach works great if axes on the scatterplot display the same unit and the range of values because then it does not matter if we go from the center horizontally along the x-axis or vertically along the y-axis, the distance to the circle contour line will always stay the same, i.e., it will be equal to the radius of the circle. For instance, if we know that the data values on the x-axis and y-axis are provided in meters and that values on both axes are in the range from 0 to 100, we know that if we create a circular brush with a radius of 10 meters, the brush will select all data points within ten meters of the brush center. In this case, the circular shape makes a lot of sense. If the scatterplot axes are scaled differently, the distance from the center of the brush to its contour lines along the x-axis and y-axis will not be the same. Figure 3.13 shows both cases. In Figure 3.13(a) We have the same quantity and axis scale. In Figure 3.13(b), we have the same quantity but a different axis scale. Since there are two radii values (one for the x-dimension and one for the y-dimension), the circle we see on the right is actually an ellipse. The view aspect ratio is 1 in both views—the view scale axis scaling, but it is essential to mention it here because this ensures that the drawn circles are not stretched due to view distortion. Our example communicates is that because of the circular shape, we are learned to think that all points on the circle are equally far away from the circle center. As our example shows, this is not necessarily true. This raises the question of when the circular brush makes sense. How is it to be interpreted if the scaling on the axes is not the same and, above all, if the data has different units. From a mathematical point of view, everything is straightforward. Instead of using the distance equation for a circle, we use the equation for an ellipse. With this, we can ensure that the display of the brush shape does not become elliptical but remains circular, even in cases when the ranges of values displayed on the two data axes are different. The distance between the point  $p = (p_x, p_y)$  and the center of the ellipse  $c = (c_x, c_y)$  is obtained with the following formula:

$$ED_{p,c} = \frac{(p_x - c_x)^2}{a^2} + \frac{(p_y - c_y)^2}{b^2} \quad (3.2)$$

where,  $a$ ,  $b$  are the radius on the x and y axes respectively. Note that radius  $a$  and radius  $b$  can have different values in data coordinates due to different scalings on the data axes. If the values are converted from data space to view space coordinates to display the brush shape, radius  $a$  and  $b$  have the same view space value, and the shape is interpreted as circular. The user must keep this in mind when interpreting the brushed data.

The implementation of percentile brushes (rectangular, square, and circular) follows a similar strategy for selecting subsets of data items, i.e., selecting a percentage of data items influenced by their dispersion from the current mouse position. Dispersion means the degree to which the distribution is compressed or stretched around the center of the

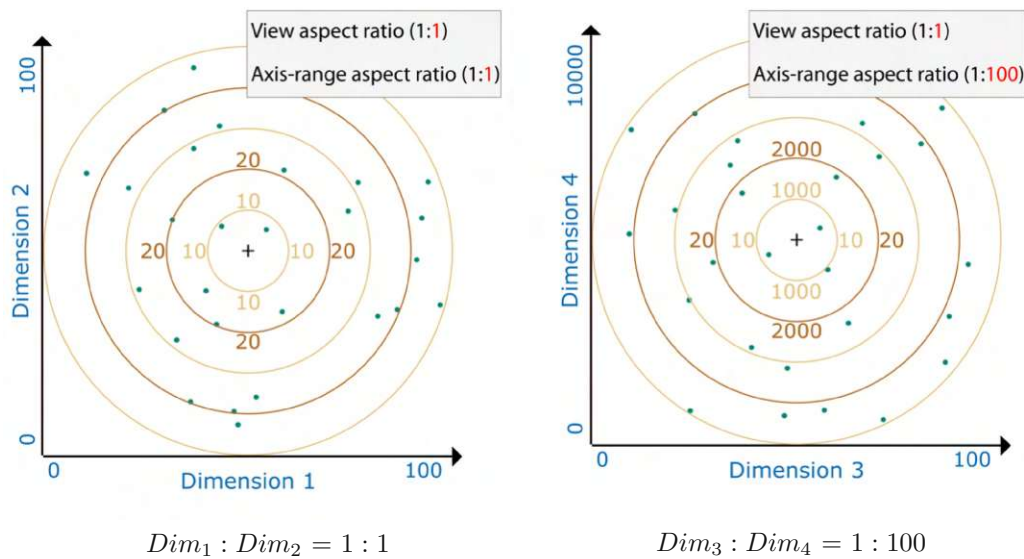


Figure 3.13: Using a circle to represent the distance from the center point. Circle contours are displayed to show a distance pattern around the center point that is rendered as a crosshair. All points on a circle are equidistant from the center. Humans are used to observing the distance to an object in this way. The effect of axis scaling on the interpretation of a circle is shown on the right.

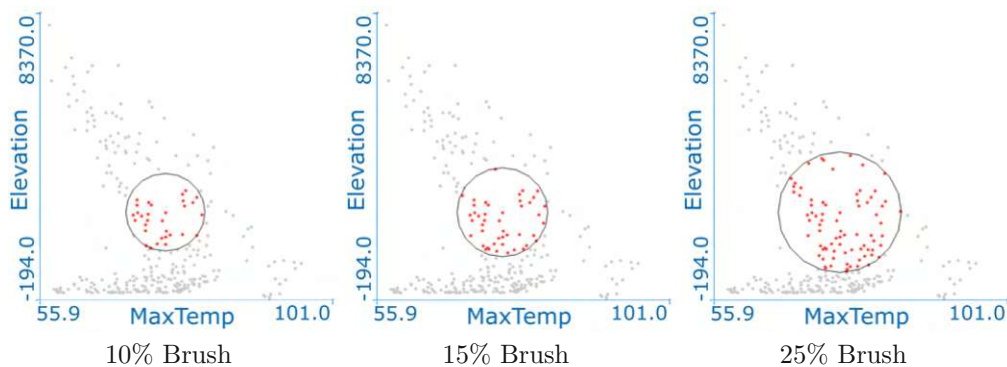


Figure 3.14: The circular percentile brush in a scatterplot. The user wants his brush to stay in the same position and is interested in selecting 10%, 15%, and 25% of all data items near the selected point on the scatterplot. After placing the brush at the desired position, he left the brush in place and began to change the percentage value. We use three different figures to display the same brush whose anchoring point stays fixed, but the radius of the brush is automatically enlarged depending on the data distribution below the brush and the current percentage value.

percentile brush. The percentile brush algorithm counts the number of items found, so the smaller the distance between items of the selected data subset on the numerical axis

(meaning their variance is low), the smaller the extent of the percentile brush. In this case of a circular percentile brush, selected data items from both data dimensions affect the radius value. Suppose the data elements in one dimension are very dense near the center of the brush, but the data in the other dimension of the data are widely scattered. In that case, the radius value increases under the influence of more significant variance in scattered distribution until the circular percentile brush has the desired percentage of data items. The brush must expand its range even more if there is significant variance within the values in both data dimensions. Conversely, if we have a dense distribution of data items in both data dimensions, the brush size will decrease because the desired percentage of data that the brush needs to select is close to the center of the brush. As with the rectangular percentile brush, the user can adjust the percentage value from 1 to 100 percent.

The primary purpose of the circular brush is to select the desired number of data items near the center of the brush. An exact range is less important than getting the desired percentage of all the data items near to the brush center, such as the closest 10% from the temperature readings depending on elevation around the brush center. The brushed range on the axes can be narrow or wide, depending on the data distribution. The user does not specify the area, just the number of data items to select with the brush. Figure 3.14 shows the effect of changing the percentage value on the extent of the circular percentile brush. In the case shown, the brush anchoring is never changed. The brush is anchored in the scatterplot of Figure 3.14(a) and it selects 10% of all data items in the scatterplot. The user can select another number of data items by changing the percentage value. The middle scatterplot shows the same brush, but here it is slightly enlarged after increasing the percentage value from 10% to 15%. The (c) scatterplot shows, as expected, that the circumference of the brush is even larger after increasing the percentage value from 15% to 25%. Conversely, assuming the brush is anchored in the same position all the time, decreasing the percentage would result in a decreasing circumference of the brush. In the case shown, the size of the brush shape changes almost proportionally with increasing the percentage parameter, so we can conclude that the spread of the 25% data items closest to the brush center is very similar.

The user can safely compare areas of two circular percentile brushes—which select the same percentage of data but are positioned at different locations—to conclude the distribution of data below the brush in relation to the mouse pointer’s position. The user should be careful when comparing brushes that select different percentages of data and are not anchored in the same place. The size of the percentile brushes depends on the data distribution, even a smaller circle can select much more data than a brush with a larger circle and lower percentage value. In practice, it will probably rarely be necessary to compare two circular percentile brushes based on the size of the circles, we would like to warn the user to be extra careful when using circles in visualizations as a comparison tool. We mean possible user misjudgment of the circle size if an incorrect mapping is used. The advice is to adjust the area of circles to match the data instead of adjusting the diameter of circles [LMvW10].



Interaction with the circular percentile brush is provided by using the mouse pointer to anchor the brush and move the brush. The user is given complete freedom in positioning and moving the brush in a scatterplot, just as with a conventional circular brush. The peculiarity of the circular percentile brush is that it automatically adjusts its extent, always selecting the same number of data items when it is moved, as shown in Figure 3.12. The center of the percentile brush can be snapped to the grid for constrained positioning. An example is shown in the scatterplot in Figure 3.15 which displays the 10% grid enabled for both data dimensions. If the snap-to-grid option is enabled and the user moves the circular percentile brush snapped to grid vertices, the brush will always jump to the vertex closest to the mouse pointer. This differs from the unconstrained movement of the circular percentile brush, where the center of the brush is placed in the mouse pointer's position. An unconstrained circular percentile brush continuously follows the movement of the mouse pointer.

To help the user move the brush snapped to the grid, we provide a brush handle. It is a line connecting the center of the currently used circular percentile brush and the mouse pointer. If the user moves the mouse pointer further away from the vertex where the brush is snapped, the brush does not immediately move with the mouse pointer, but the line handle is extended to create a visual connection between the mouse pointer and the active brush. As soon as the distance to another vertex of the grid is shorter than the distance to the current one, the brush will jump to that other vertex, which will be the new center point of the brush (this case is illustrated in the scatterplot in Figure 3.15). From the feedback given during the demonstration session, we found that the auxiliary brush handle is not necessary if the grid cells are prominent, i.e., well visible, and if the vertices are at a sufficient distance so that the user can visually conclude which vertex is closest to the mouse position, as is the case we have shown. If the grid is divided into many cells, as is the bottom central part of our 10th percentile grid, the line handle is helpful for users because it unambiguously points to the vertex where the brush is snapped to.

The circular percentile brushes can also be combined using logical operators and expressions like for traditional brushes to create a more (complex) composite brush. Tools that provide interactive brushing in CMVs commonly provide logical AND and OR operators. The AND operator allows the refinement of the brushed data subset by creating additional brushes in the linked views, while the OR operator is helpful if the user wants to include additional data items to the brushed data subset (for more details on composite brushing, see Martin and Ward [MW95]). For instance, let us say that we are interested only in 15% percent of all data items in a scatterplot whose values fall between the 5th and 20th percentiles of the data in the vicinity of the user-selected point. We can create an appropriate selection using two circular percentile brushes, as illustrated in the scatterplot of Figure 3.14. Using unconstrained brushes, positioning two circular brushes in the scatterplot so that their centers are in the same position is difficult. Using the snap-to-brush option makes this task quick and accurate. We combined two circular percentile brushes in a single composite brush by stacking them directly on top of each



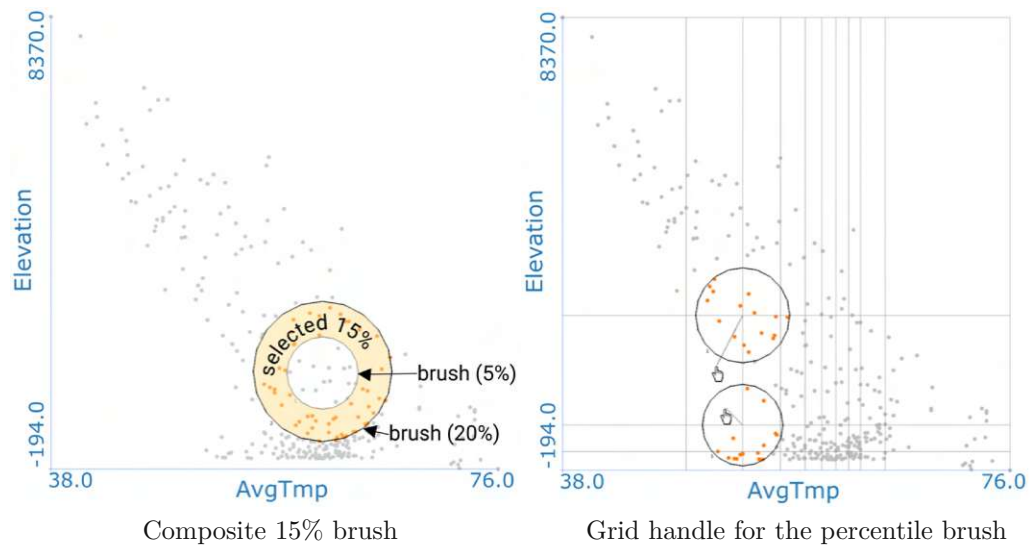


Figure 3.15: **a**: A scatterplot with a (smaller) 5% circular percentile brush and a (larger) 20% circular percentile brush. Both brushes are placed at the same position (they have the same center point), and their size is adjusted automatically. Exact placement was easy by using the snap-to-brush option. A composite brush was created (using logical XOR operator) to select all data items that are in the larger brush but not in the smaller one. **b**: The auxiliary handle, presented as a line running from the center of the brush to the mouse pointer, helps the user move the circular percentile brush, whose movement is constrained by the snap-to-grid brush option. Two brush positions are shown. Notice based on the additional line shown which vertex is closest to the mouse pointer's position.

other. We use the logical XOR operator, an exclusive disjunction that is true if and only if its arguments differ. Such rank-based analysis, enabled with percentile brushes, can be very useful for users who want to do quantitative analysis. Once the circular percentile brush is created, the user will probably want to continue the quantitative analysis of the brushed subset in other data dimensions visualized in the linked views.

### 3.6 Mahalanobis Brush

Interactive visual analysis has long lacked a brushing technique that is easy to use and can quickly brush elongated (elliptical) structures visualized in a scatterplot, such as *Subset A* displayed in the scatterplot in Figure 3.16(a). Having a brush that leverages information about the underlying data distribution to modify its shape and possesses the capability to adjust its rotation in relation to the data would be beneficial. We briefly discuss the advantages and disadvantages of the several brushing techniques in the context of our test case. Brushing scatterplots traditionally involves simple brush geometries, such as a rectangular brush and a circular brush, as well as a lasso tool that allows users to create arbitrary geometries. Figure 3.16 provides a comparison of several

different brushing techniques for selecting *Subset A*, the problematic data distribution that is stretched and not parallel to the scatterplot axes.

Simple brush shapes, such as rectangles and circles have the advantage of being fast to create, but their use is challenging if one wants to brush an elliptical and rotated distribution. Due to its rounded shape, the circular brush is not suitable for selecting elongated distributions. The length of the distribution determines the circle's diameter, and therefore the area of the circle will be relatively large in relation to the data we want to brush. Consequently, there may be data items on both sides of the distribution selected with the brush in addition to those we intend to select.

A rectangular brush shape is suitable for relatively quickly and accurately selecting elliptical distributions parallel to the axes of a scatterplot. Accuracy is lower if such a distribution is rotated. To select all data items from *Subset A* the extent of the rectangular brush was made very large both vertically and horizontally, as shown in Figure 3.16(b). Consequently, some data items that do not belong to the distribution are also included in the selection.

The advantage of working with a rectangular brush shape parallel to the coordinate axes is that it is easy for the user to describe it mathematically. Because the edges are parallel to the axes, the selected interval is defined by the vertices of the rectangle. If we move the rectangular brush or change its extent, we change the brushed interval on the corresponding axis, and the brush selects all data items that belong to the new interval. Some tools offer a rotation of a rectangular brush. If the sides of a rectangular brush are not parallel to the axes of a scatterplot, the user has to exert a higher cognitive effort to explain such a brush quantitatively and to form a mental image of the range of values being brushed.

The technique known as composite brushing [MW95] allows users to define their focus more specifically by assembling a complex selection from multiple individual brushes using logical operators. Figure 3.16(c) shows a composite brush consisting of four rectangular brushes combined with a logical OR operator. While adjusting the extent of individual rectangular brushes takes time, this approach achieves high brushing accuracy even when using simple geometries, such as rectangles. Such complex compositions are difficult to understand, not easy to reproduce, and clutter the visualization.

With percentile brushes, we have transitioned from traditional screen space brushing to a method that considers the underlying data during the selection process. We only investigated the combination of rectangular, and circular brush shapes with data distribution information. The square brush changes its extent but remains a square. The circular brush changes its radius but remains circular. Both brushes failed in terms of being accurate at selecting *Subset A*. The square percentile brush is shown in Figure 3.16(d), and the circular percentile brush is given in Figure 3.16(e). For the reasons explained above, some other data items not belonging to *Subset A* were selected.

The Lasso brush has proven to be the most successful for the task at hand. The advantage of the lassoing technique is that it allows users to select precisely the subset of data they

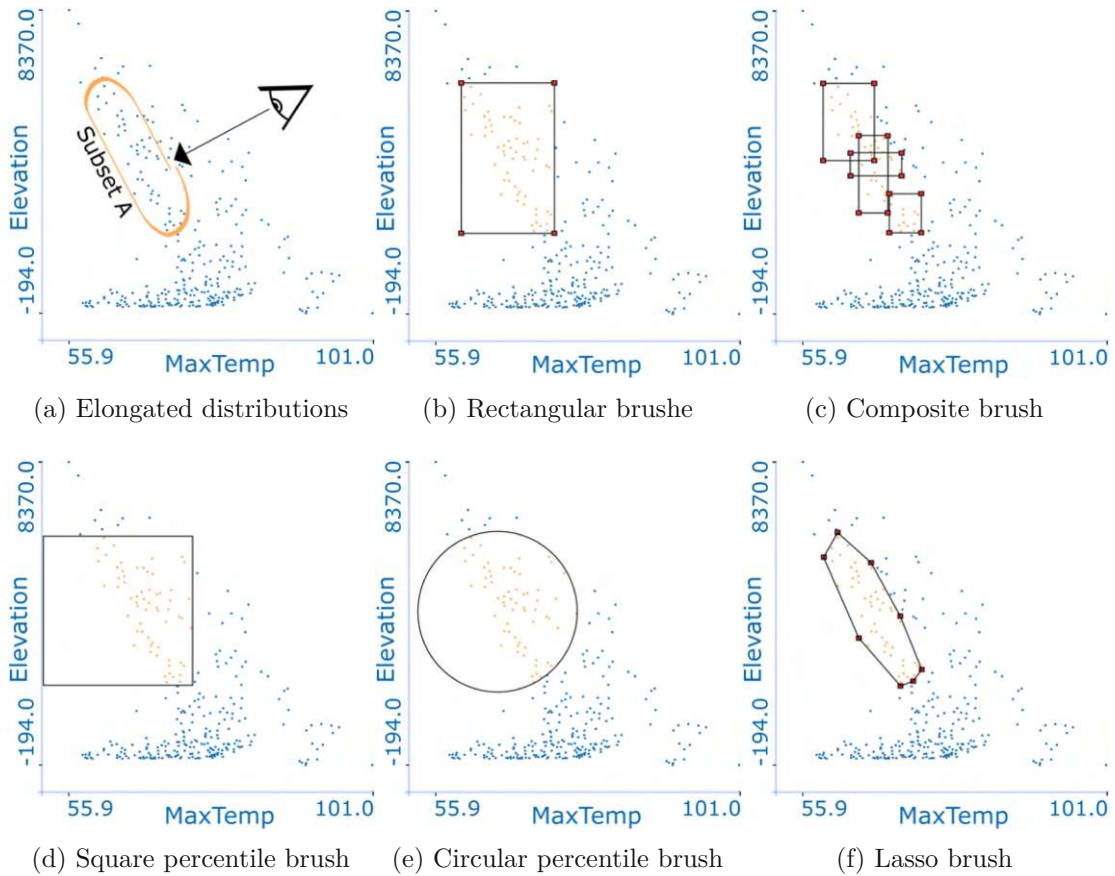


Figure 3.16: **(a)** The elongated and rotated data distribution (*Subset A*) is highlighted in orange for clarity. Five different brushing techniques were tested to select this distribution. **(b)** The rectangular brush fails in terms of being precise. **(c)** The composite brushing technique allows for relatively precise selections, but is demanding to create. **(d)(e)** Both the square and circular 20% percentile brushes successfully selected *Subset A*, but also some additional data items that do not belong to this subset. **(f)** By specifying a lasso brush, point by point, it becomes possible to accurately select all relevant data items.

intend to analyze. The user creates a lasso brush by interactively sketching a detailed geometric shape around an area or data items of interest in a scatterplot. As Figure 3.16(f) demonstrates, with a lasso brush, we precisely selected the rotated distribution, a task that proved challenging with other traditional brushing techniques. However, the lasso brush has several disadvantages. First, a lasso brush can be time-consuming to create, which also negatively affects the fluidity of the visual analysis process. The second disadvantage is its complexity both in terms of mathematical computation and interpretability. If the geometry of a lasso brush is complex, it can be problematic for the user to describe such a brush or move it around in a scatterplot. In fact, only in rare cases does it make sense to move a lasso brush. The third disadvantage is that it is not easy to reproduce a lasso brush that has a complex shape.

The results from our small evaluation reveal that all tested brushing techniques have managed to brush *Subset A*, i.e., the rotated and elongated data distribution. The lassoing and composite brushing techniques are accurate but slow. Simple brush shapes are fast at selecting but inaccurate. The selected data subset includes data items which do not belong to the subset we are interested in, and it would be good to get rid of them automatically. A critical part of an automatic brush that could help us here would be the ability to identify the data items belonging to the underlying distribution.

Clearly, the data that make up a distribution are in a specific correlation, which is not the case concerning the surrounding data that are not part of that distribution. Statisticians have developed various techniques for calculating the distance of a specific value from the distribution. The Euclidean distance is the most common measure of distance in everyday life, for example, the distance between two points in Euclidean space is the length of a line segment between the two points. We can utilize the Euclidean distance to measure the distance of a point from the center of a subset of points in a scatterplot. In Euclidean space, the variables are represented by axes drawn at right angles. The use of the Euclidean distance in a scatterplot results in evenly spaced circles around the center point, as displayed in Figure 3.17(b).

We use a scatterplot to display data dimensions Elevation and MaxTemp in our example. Because the scales of the measurements are different, the resulting distances are skewed based on the units. Due to the variations in units used for each variable, it is essential to effectively address these differences if we are interested in data distribution information. We would like to scale the distance, i.e., to include variance in the distance calculation. The first variable Elevation has a more significant inter-sample difference, i.e., it exhibits a notably larger range of values between the samples, so its contribution to the distance calculation will be higher.

We must balance out the contributions, i.e., to standardize data that eliminates units and weighs both measures equally. To accomplish this, we can adjust the Euclidean distance equation (Equation 3.2) to incorporate variance information from the data.

The scaled Euclidean distance equation (Equation 3.3), as defined in Deza et al. [DD09],

provides a solution:

$$SED_{p,c} = \sqrt{\frac{(p_x - c_x)^2}{\sigma_1^2} + \frac{(p_y - c_y)^2}{\sigma_2^2}} \quad (3.3)$$

where  $\sigma_1^2$  and  $\sigma_2^2$  are the variances of the data in the horizontal and vertical dimensions, respectively. If the variances are different, circles are transformed into ellipses, as depicted in Figure 3.17 (c). With this step, we can already create a data-aware brush, i.e. to select data with partial awareness of the trend in the data. Ellipses are still oriented along the axes of the Cartesian coordinate system, i.e., they do not rotate in the direction of the data distribution.

In comparison to the scaled Euclidean distance, the Mahalanobis distance metric (MD), introduced by P.C. Mahalanobis [Mah36], has one crucial difference we want to take advantage of: it incorporates the correlation between data variables that make up the distribution. This particular aspect is captured by the covariance matrix and will allow the ellipse to rotate. The covariance matrix employed in the MD distance Equation 3.4 captures this specific aspect, enabling the ellipse to rotate accordingly. This covariance matrix retains information about whether the data items from the selected data subset have different variances and whether correlations exist among them. The following equation gives the MD distances from a point  $p = (p_x, p_y)$  to the center  $c = (c_x, c_y)$  of a distribution:

$$MD_{p,c} = \sqrt{\begin{bmatrix} p_x - c_x \\ p_y - c_y \end{bmatrix}^T C^{-1} \begin{bmatrix} p_x - c_x \\ p_y - c_y \end{bmatrix}} \quad (3.4)$$

where  $C$  is a sample covariance matrix of the data set. Multiplying by the inverse of the covariance matrix only reduces the distances for strongly correlated data values, and consequently values not belonging to the distribution are considered farther from the distribution center.

In the example, in Figure 3.17 (c), the data values of the selected subset have a negative correlation. Because the MD considers axes resulting from the data itself, the ellipses are rotated and follow the orientation of the distribution. It might be tempting to assume that the values on the elongated side of the ellipse are considerably distant from the distribution center. According to the Mahalanobis metric, this is not the case as on this ellipse, all values on the curve are equidistant from the center of the distribution.

Our work [RSM<sup>+</sup>16] was the first in visualization research to consider the use of the MD for interactive data selection in a scatterplot. The Mahalanobis brush we introduced knows whether the components from the selected distribution have different variances and whether there are correlations between the components, based on which it adjusts the shape of the brush, its orientation, size, and position. The Mahalanobis brush possesses the capability to distinguish the data items belonging to the underlying distribution while avoiding those situated near the distribution. Additionally, we want the Mahalanobis brush to work like percentile brushes, which means the user should be able to specify

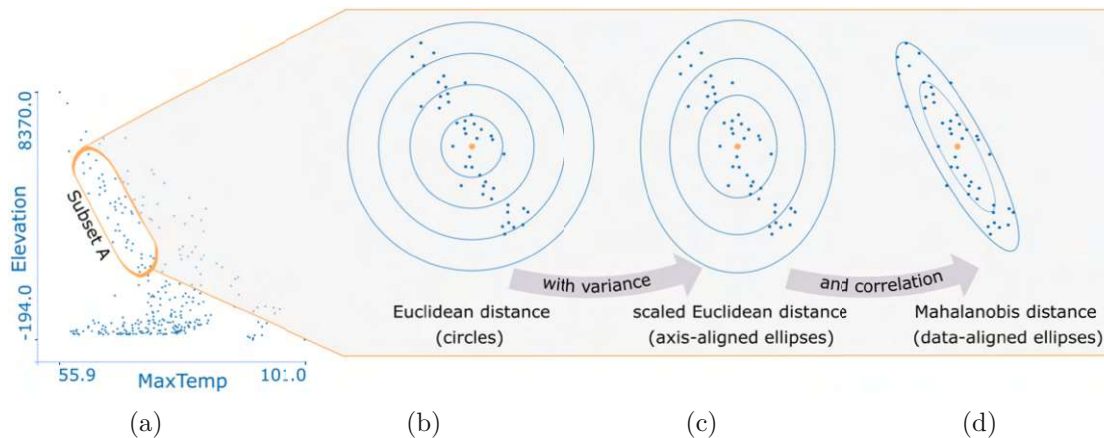


Figure 3.17: Comparison of three different distance metrics: **(b)** Euclidean, **(c)** Scaled Euclidean, and **(d)** Mahalanobis, using *Subset A* shown in **(a)**. Data points from *Subset A* are shown enlarged, together with their center (orange point). Additionally, isolines (contours) for each distance metric are displayed to show a distance pattern around the center point. For each distance metric the distance to the center is the same on each curve. The results show that the Mahalanobis distance with data-aligned ellipses provides the most desirable results for the user.

how much data to select. Because it is quantitatively interpretable, users can easily understand the Mahalanobis brush even if they are unfamiliar with the Mahalanobis distance metric.

The covariance matrix accounts for the covariance between data dimensions and the fact that variances in different data dimensions may be different. The covariance matrix values directly influence the shape of the Mahalanobis brush. Table 3.2 illustrates the link between the covariance matrix and the appearance of the (Mahalanobis) ellipse. The three examples of covariance matrices on the left have zeros in off-diagonals. Since there is no correlation, ellipse axes are parallel to coordinate the axes. In the remaining four matrices, non-zero values off-diagonally indicate that the data has some variance that is not aligned with the axes. In the two middle matrices show that when the two variables are negatively correlated, the covariance has a negative value, and the corresponding ellipse rotates  $-45$  degrees. The two matrices on the right show an example of positively correlated variables, and the corresponding ellipse rotates  $45$  degrees. In the case of a perfect correlation, instead of an ellipse, there will be a line.

Eigenvectors (principal components) calculated from the covariance matrix are associated with the analyzed data, and we can use the two most prominent principal component directions to form a rotating coordinate frame. The axes of the ellipse are adjusted automatically such that the major axis of the ellipse points in the direction in which the data varies the most (corresponds to the first principal component which is given by the



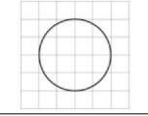
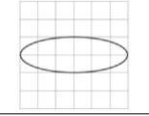
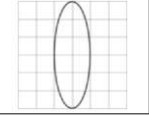
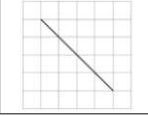
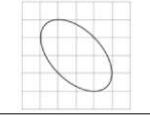
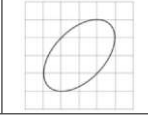
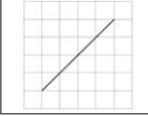
$\begin{bmatrix} 4 & 0 \\ 0 & 4 \end{bmatrix}$	$\begin{bmatrix} 9 & 0 \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 \\ 0 & 9 \end{bmatrix}$	$\begin{bmatrix} 4 & -4 \\ -4 & 4 \end{bmatrix}$	$\begin{bmatrix} 4 & -2 \\ -2 & 4 \end{bmatrix}$	$\begin{bmatrix} 4 & 2 \\ 2 & 4 \end{bmatrix}$	$\begin{bmatrix} 4 & 4 \\ 4 & 4 \end{bmatrix}$
						

Table 3.2: Representation of a 2x2 covariance matrices by ellipses. The covariance matrices can have different values depending on the shape of the data (we show only the resulting covariance matrix and the corresponding ellipse, not the data values themselves).

eigenvector with the largest eigenvalue), and the minor axis of the ellipse is orthogonal to the major axis (corresponds to the second principal component which is given by the eigenvector of most significant variance among those that are orthogonal to the first eigenvector). Ellipse will be elongated according to how much the data varies along the respective dimension. In case that variances in both dimensions are equal the ellipse is a circle. Assuming there is no correlation, the covariance matrix in two dimensions contains only horizontal and vertical variances. If we include such a sparse matrix (zero correlation in off-diagonal entries) in the Equation 3.4, for the Mahalanobis distance: the MD equation is reduced to the equation for the scaled Euclidean distance (compare Equation 3.5 and Equation 3.3).

$$\begin{aligned}
 MD_{(p,c)} &= \sqrt{\begin{bmatrix} p_x - c_x & p_y - c_y \end{bmatrix} \begin{bmatrix} \frac{1}{\sigma_1^2} & 0 \\ 0 & \frac{1}{\sigma_2^2} \end{bmatrix} \begin{bmatrix} p_x - c_x \\ p_y - c_y \end{bmatrix}} \\
 &= \sqrt{\begin{bmatrix} \frac{p_x - c_x}{\sigma_1^2} & \frac{p_y - c_y}{\sigma_2^2} \end{bmatrix} \begin{bmatrix} p_x - c_x \\ p_y - c_y \end{bmatrix}} \\
 &= \sqrt{\frac{(p_x - c_x)^2}{\sigma_1^2} + \frac{(p_y - c_y)^2}{\sigma_2^2}}
 \end{aligned} \tag{3.5}$$

Now that we have explained why the ellipse changes shape based on the underlying data, we briefly explain the key steps to create the Mahalanobis brush. These steps are given in Figure 3.18. The axes of a scatterplot are not shown to provide more space for texts in the figures (for the reader’s reference, it is the same data as displayed in the scatterplot in Figure 3.16). The main steps for computing the Mahalanobis brush are also given in Algorithm 3.1.



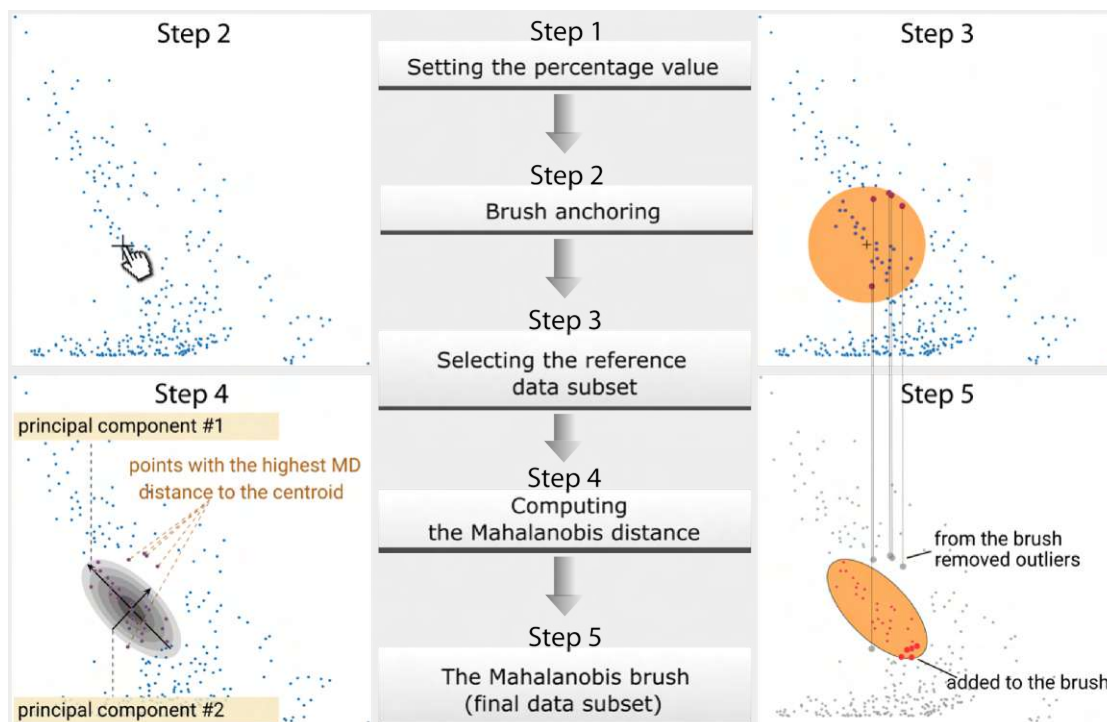


Figure 3.18: A schematic overview of the steps to create the Mahalanobis brush technique in scatterplots.

**Step 1: Setting the percentage value.** The percentage value assigned to the percentage parameter  $n$  quantitatively characterizes the Mahalanobis brush and represents the proportion of data items selected from the entire dataset. By changing the percentage value, users express their desire for a certain amount of data that the Mahalanobis brush should select, and the brush is updated automatically. When creating the brush, the user does not have to set the percentage value if he is satisfied with the default value (set to 10%). He can return to this step at any time and change the percentage value as desired, from 1% to 100%. Knowing the percentage value, the user can quantitatively interpret the Mahalanobis brush even when it is moving through the scatterplot and its shape (size and orientation) changes depending on the underlying data. That is in line with the previously presented percentile brushes, allowing rank-based analysis. Alongside the percentage parameter, we introduced the sensitivity parameter  $m_d$ , which serves to alleviate the influence of minor cursor position adjustments while moving the Mahalanobis brush. With this parameter, users can specify how many data items from the local distribution the Mahalanobis brush will consider for orientation calculation. Using a smaller value increases the sensitivity of the Mahalanobis brush to changes in the immediate vicinity of the cursor position. **Step 2: Brush anchoring.** By clicking the left mouse button on a scatterplot near the distribution center to be selected with the Mahalanobis brush, the user initiates the creation of the brush. The click position

does not necessarily correspond to the calculated anchor, i.e., center of the distribution selected by the Mahalanobis brush, but it instructs the automation in the third step to compile a subset of the reference data  $D$ , based on which the anchor point is determined. The anchor point is set in the fourth step after outliers from the reference data subset are removed. For the anchoring, the arithmetic mean of the reference data subset is used, i.e., the brush is translated such that its center is at the centroid of the distribution under the brush. The anchor point of the Mahalanobis brush is also used in the fifth step when drawing an ellipse. **Step 3: Selecting the reference data subset.** Here, an initial data subset  $D$  is created that contains a specified number of data items near the mouse cursor position. The exact number of data items to be in this reference subset  $D$  is obtained from the percentage value. Choosing the right data items for the subset  $D$  is important because, in the fourth step, we calculate the covariance matrix from this subset, and it ultimately affects the accuracy of the final brush selection. To demonstrate our idea, we do not optimize this step in a sense of making optimized selection of the local context for the Mahalanobis computation, and rely on selecting a desired number of data items by using the percentile brush technique presented earlier in this chapter. This method is fast and has proven to be accurate enough for the practical application of the Mahalanobis brush. The upper-right scatterplot in Figure 3.18 shows the circular percentile brush that is automatically scaled to select 10 percent of all data items closest to the mouse cursor (note that visualizations in steps three and four are displayed only for the demonstration purpose; the user sees only the final Mahalanobis brush, as shown in the scatterplot on the lower right). Our decision to use a circular shape to select the reference data subset is supported by the fact that when there is no correlation in the data and variances are equal, the ellipse representing the Mahalanobis brush turns into a circle. In such a case, the reference subset  $D$  will be equal to the final brushed subset, meaning that the overall calculation will be faster. The distribution that we want to select here is elongated and rotated, and as explained at the beginning of this section, using the circular shape for selecting the elongated distribution can potentially result in outliers being selected. By default, the values for the  $n$  and  $m_d$  parameters are the same, so  $m_d$  does not affect the brush. However, the user can change the value of the  $m_d$  parameter and directly adjust the size of the reference subsets  $D$ . For example, if the user wants to select a very elongated distribution and the Mahalanobis brush selects some unwanted data items that are not part of the distribution but are close to it on the sides, he can reduce the area considered for the calculation of the covariance matrix, to lengthen the ellipse without reducing the total number of data items the Mahalanobis brush needs to select. A better way that we have not implemented would be to iteratively refine the reference subsets  $D$  by replacing data values whose Mahalanobis distance is larger than the newly considered neighbor values up to a reasonable convergence. **Step 4: Computing the Mahalanobis distance.** In this step the most important operations take place. The covariance matrix is computed, rotation of the brush ellipse is determined, and the final anchor point of the brush is set. These are the elements needed to define the Mahalanobis brush. These can be obtained from the reference data subset  $D$ . After computing the covariance matrix from the reference subset  $D$

we clean up this subset by removing the outliers (using the Mahalanobis distance to estimate which data items are outliers, as described at the beginning of this section, see Equation 3.4), and then use the updated subset  $D$  to calculate the anchor point and the final covariance matrix from which we extract the first and second principal components for ellipse generation. Calculating eigenvectors and eigenvalues in the case of a 2x2 covariance matrix involves finding the determinant of the matrix. Since the determinant of a 2x2 matrix is a polynomial of degree 2, it can be factorized and solved using explicit algebraic formulas. **Step 5: Finalizing the Mahalanobis brush.** The brushed subset of data items is highlighted, and the ellipse is drawn around that subset, i.e., the Mahalanobis brush is displayed and the user can now interact with the brush.

---

**Algorithm 3.1:** Mahalanobis brush.
 

---

**Data:** all data in the horizontal and vertical dimension,  $\vec{p}$ : mouse position, percentage  $n$ : of all data items to be brushed, percentage  $m_d$ : all data items forming the basis for computing  $C$ ,  $d$ : data items closest to point  $\vec{p}$

**Result:**  $M$ : all brushed data items

```

/* Step 1: Computing the Mahalanobis metric. */
1 while percentage of data items in the subset  $D < m_d$  do
2   | increase size of subset  $D$  by adding nearby data items;
3 end
4  $C \leftarrow \text{ComputeCovarianceMatrix}(D)$ ;
5  $d \leftarrow \text{ComputeMahalanobisDistances}(\vec{p}, D, C, n)$ ;
/* Step 2: Aspect ratio and rotation of the brush ellipse
   according to an eigen-analysis of  $d$ . */
6 while percentage of data items selected by the brush  $< n$  do
7   | increase the size of the ellipse depending on the variance of  $m_d$  and
   | associate the contained data items with the selected subset  $M$ ;
8 end

```

---

The Mahalanobis brush responds instantly to all user interactions by updating its shape and position. The user can move the Mahalanobis brush freely like other traditional brushes in the scatterplot. If moved, the Mahalanobis brush automatically adjusts its shape, size, and orientation but always keeps the predefined number of selected data items. The Mahalanobis brush supports a quantitative analysis. By positioning the mouse cursor and/or altering the percentage parameters, users can more easily select elongated and rotated distributions given in a scatterplot. Because the orientation and elongation of the Mahalanobis brush change depending on the underlying data, the user gets additional information about correlation and variance in the data. For example, the ellipse with positive covariance rotates to the right, which means that the values of the first and second data dimensions both increase. Figure 3.19 shows how the Mahalanobis brush changes its shape when moved in a scatterplot.

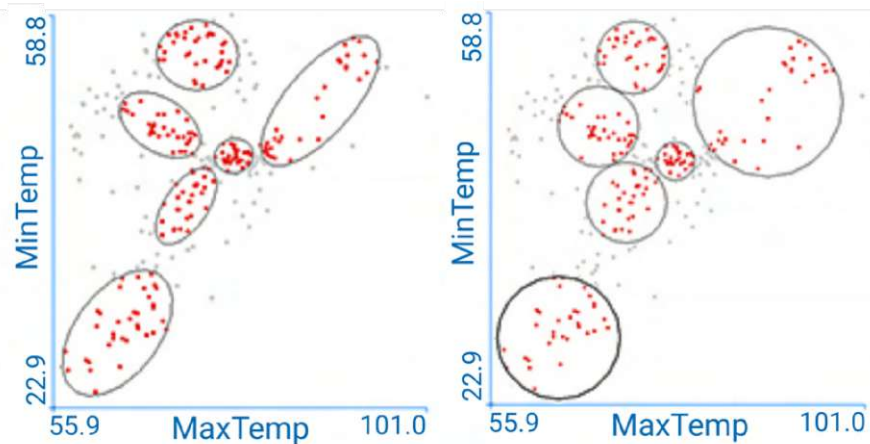


Figure 3.19: The 10% Mahalanobis and 10% circular brushes are displayed. By relying on the Mahalanobis distance metric, the Mahalanobis brush reduces the area covered by the brush so that it selects less outliers (considering the data under the brush) than the circular percentile brush.

### 3.7 Animated Brushing

The analysis process in general consists of a series of brushing actions performed by the user—including positioning, resizing, and moving brushes—and system responses that come as reactions to each user’s actions. Those system responses typically manifest themselves as changes in visualizations. The changes that occur over time in the linked views can be complex and occur in different ways, which the user may find easier or harder to understand. The success of the interactive visual data analysis process may depend on how successfully users interpret related visualization updates. To reduce users’ mental burden in interpreting changes in the linked views, we propose *animated brushing*. In addition, this technique can increase the accuracy and support the reproducibility of the brushing operations.

Animated brushing is a controlled animated sequence of individual brushing operations that the user can control using standard commands familiar from video interaction, including start, pause, play, stop, and replay. Each animation frame represents an actual brush operation in the brushed view. The distinction from traditional brushing, where the user manually controls the brush, lies in using an animation framework. This framework allows us to automatically execute brushing operations step by step (frame-by-frame) in the form of an animation, a process known as animated brushing.

Users often find themselves in a situation where their attention cannot be primarily focused on the changes occurring in the linked views because they are busy taking care that the brush is moving precisely along some desired path. Sometimes even a slight deviation of the brush placement in the brushed view causes significant changes in relations between the brushed data items in the linked views. In such situations, to

understand the critical changes in the linked views, users must repeat the brush movement in the same way several times. Difficulties in reproducing the path of the brush moving in the scatterplot are already discussed in Section 3.1 and illustrated in Figure 3.1. An animated brush can be highly beneficial in this context. On one hand, in the brushed view, animating the initialization and resizing of the brush ensures the reproducibility of these operations. Additionally, users gain the ability to stabilize the brush movement, and repeat a sequence of brushing operations. On the other hand, in the linked views, the user can focus on comprehending the changes in the visualized data.

Figure 3.20(a) demonstrates animated brushing in a scatterplot. Figures 3.20(b-d) provide an example of a basic configuration for animation controls. The user can define the start and end frame by placing a brush at two distinct positions in a view or entering values for an exact positioning via the GUI interface. The user presses the start button to initiate the animated brushing in the brushed view, causing the brush to move and adjust its shape according to the configured settings. In addition to enabling controlled brush movement, the user can set the total animation time. In each animation frame, the subset of brushed data is also highlighted in the linked views through the basic concept of linking&brushing.

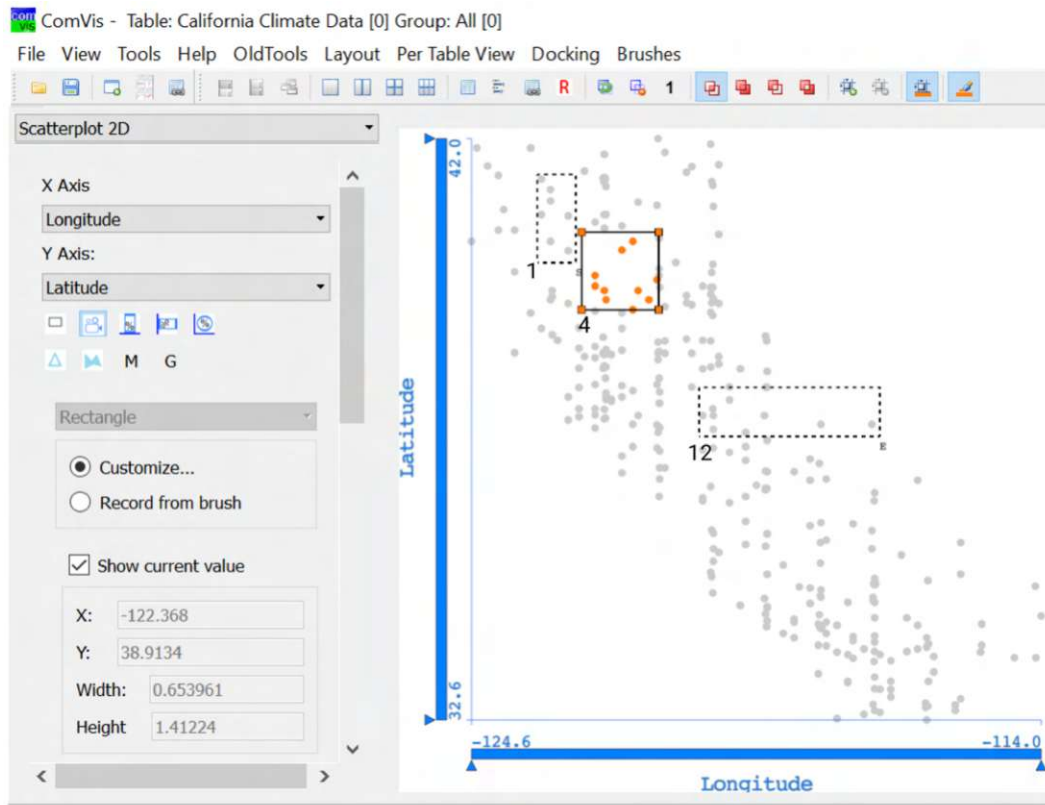
We considered two different methods for recording animated brushing in a scatterplot, but these can also be applied in other visualizations. The first method generates animation frames based on the user’s interactive brushing operations within a view, and the second method is a controlled approach (automatic or semi-automatic) for producing a sequence of animation frames.

Interactive brushing can include unconstrained brushes and semi-constrained brushes limited in any of three aspects: anchoring, range, and movement. The recorded animation of unconstrained brushes can result in a huge number of frames, of which we might only need a few because there has not been any change in the selected data subset. An update of the linked views is usually done only when the brushed subset changes. If the purpose of utilizing animated brushing is to facilitate the observation of related changes across linked views, we can get rid of adjacent frames that do not capture changes in the brushed data.

We do not save changes such as, for example, minor changes in brush position due to the user’s unsteady hand if such a change has not resulted in an update of the selected data subset. A new animation frame is saved only if the subset of the selected data changes compared to the previous frame. An exception to this rule exists: If our animation system detects that the user is utilizing constrained brushing, each brush operation automatically translates into a new animation frame. For instance, if a brush movement is constrained to grid vertices, as discussed in Section 3.2, the brush is shifted from one cell to another, and we add a new frame for each cell. This automatic detection of brushing operations can be turned off by the user if desired. Furthermore, our animation framework supports manually deleting irrelevant frames for the user.

The second option for animated brushing is to generate (brushes) frames automatically

### 3. REPRODUCIBLE BRUSHING



(a)

(b)

(c)

(d)

Figure 3.20: (a) Animated brushing in a scatterplot. The current frame of a recorded 12-frame animation is shown with thick lines and the starting and ending frames with dashed lines. Once the path for an animated brush is defined, users can control the animated brush using standard animation commands such as start, pause, and play, demonstrated in (b-d). The animated brush updates as the animation progresses.



or semi-automatically. There are many ways how this can be technically achieved. The process of creating an animated brush in a scatterplot using a semi-automatcal approach is schematically displayed in Figure 3.21. We have implemented a solution where the user defines a total number of frames to be generated, including the start and end frame of the animation. The animation framework automatically generates additional (in-between) frames by linearly interpolating between the start and the end frame, as illustrated in Figure 3.21(b). It is often the case that the user moves the brush in a straight line. The possibility of linear interpolation helps the user understand the changes related to the brush's movement more easily.

An interesting characteristic of our approach is the additional interactivity. We follow the advice of Tversky and Morrison [TMB02] and include interaction in the animation to stimulate interest and further encourage users to explore the visualization. Once the frames for the animated brushing are recorded, the user can visualize and adjust them. Each frame represents a unique brush and has information about the geometric properties of the brush.

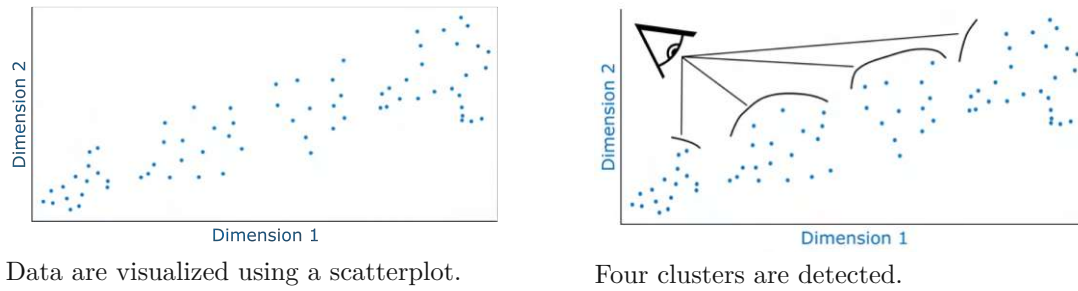
We only store data related to the creation of the brush. For example, if the shape is a circle, we will store solely the center point and radius, which is the minimal data required to reproduce the brush. The brushed data is automatically generated when reproducing a brush.

When we discuss interacting with an animation frame, we mean modifying the attributes of the associated brush. As an example, the user can adjust the anchoring and extent of the animated brush, and subsequently start/continue the animated brushing.

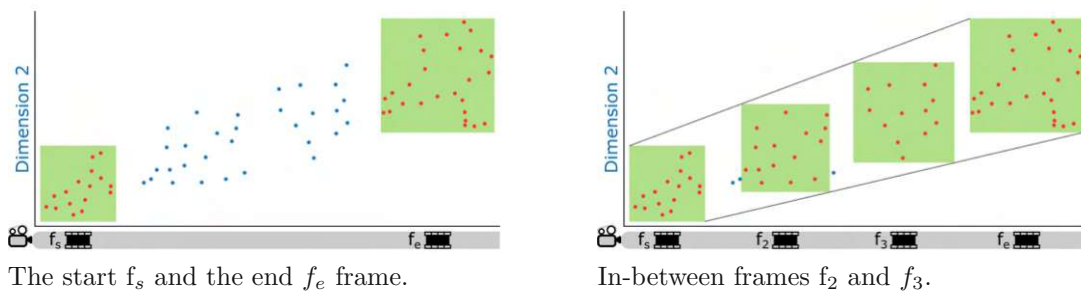
Figure 3.21c displays an example of updating the second frame by changing the brush extent to include three additional data items. The user can choose whether the changes related to the selected data are postponed until the editing of animation frames is completed or, as with a traditional brush, the visualizations in linked views are updated immediately. In addition, the user can insert additional frames by defining them in the view or specifying them through animation controls.

Experts appreciated the possibility of editing recorded animation frames. They pointed out that animated brushing is well-suited for exploratory data analysis. This is because it provides the opportunity to return to a particular location at any time, modify a brush's shape for a specific frame, delete unimportant frames, or continue the analysis in another direction.

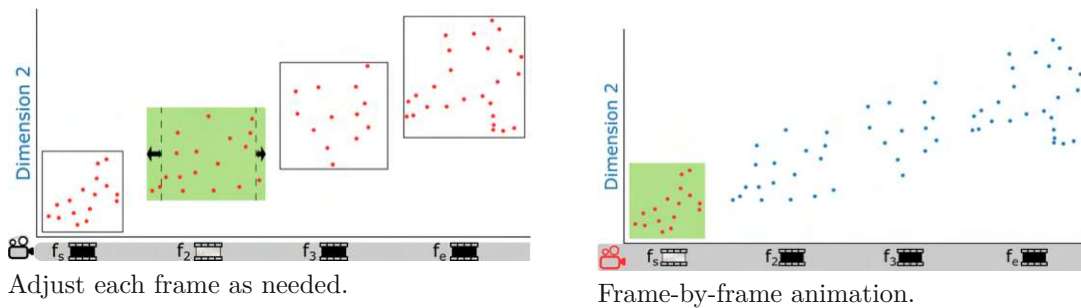
### 3. REPRODUCIBLE BRUSHING



(a) Define the analysis task



(b) Create the animated brush



(c) Start the animated brushing

Figure 3.21: A schematic representation of the steps taken to create an animated brush in a scatterplot. **(a)** Based on a visual analysis of the data, the user decides how to move the brush to select interesting subsets of the data. **(b)** The user defines where the brushing process will begin and end. If linear interpolation is chosen as the method for generating new frames, the animation framework automatically creates intermediate (in-between) frames. **(c)** If necessary, the user can adjust the shape of the brush at any time and in any position defined along the animated sequence. The animated brush moves automatically as the animation progresses.

# Quantitative Linking

This chapter addresses the need for more quantitative information in visual analysis. Section 4.1 explains the motivation behind enabling quantitative results in addition to qualitative insights from visual analysis. In Section 4.2 we introduce several extensions that support the user’s quantitative interpretation of the linked views. These extensions are specifically tuned to communicate the quantitative readings from the brushed data while maintaining the interactive and dynamic nature of the visual analysis. To help interpret changes in the linked views, Section 4.3 also introduces the relative difference plot, a novel way of describing the history of linked data statistics.

## 4.1 Motivation

Visual analysis is taking on an increasingly important role in data exploration and analysis. It supports comprehensive analysis of diverse data sets using various qualitative visuals to bring out the key characteristics of the data. Due to the large visual bandwidth of humans, visual analysis’s qualitative character naturally harmonizes with a human’s integration in the analysis loop, which is usually achieved through the use of interactive and linked visualizations. A lot of interactive visual analysis predominantly delivers qualitative results—based, for example, on a continuous color map or a detailed spatial encoding. Typically, the brushed data subset is visually highlighted, while the rest of the data set is shown as context, for example, differently colored, smaller, or accumulated [Hau05]. As the brush moves in the brushed view and to gain insight into complex correlations between different data dimensions, the user observes how visualizations change in the linked focus+context views. If changes are many or complex, in order to succeed, the user must carefully observe the data and create a mental model of the changes shown. This results in only approximate readings of such views, which is not always sufficient. The following example statement is intended to illustrate this situation: “Using linking&brushing, we see that low values of dimension x [as brushed in view A] are correlated with high

values of dimension  $y$  as apparent in the linked focus+context visualization [view B].” The meaning of “low” and “high” remains vague/relative. Computational data analysis would usually put a number on such a relation—maybe a Pearson correlation coefficient. The brushed and linked visualization also provides additional information about the relation between  $x$  and  $y$ . It indicates if the relation is linear or not, for example, and this is highly useful. If relations established through linking&brushing are complex, even with full attention to the linked view(s), additional methods are required to support understanding and quantify the analysis results. With a better understanding of what is happening on the brushing side, as described in Chapter 3, we also aim at a better understanding of the linked side.

In decision-making, hard quantitative facts are frequently invaluable, in addition to a useful, qualitative visualizations. Critical target applications of IVA, such as medical diagnosis and business intelligence tools and processes, clearly benefit from quantitative results to transform data into insight that supports decision making. Since visual analysis traditionally delivers primarily qualitative results, business analysts, for example, prefer numerically oriented tools that support the quantitative analysis of the data and can help them to make data-driven decisions more quickly [KBHP14]. Decision making is very often done based on the values provided by summary statistics. As analysts need quantitative results, and statistics can provide these, a logical step is to enhance the visual analysis with statistical values about the brushed data.

## 4.2 Inclusion of (Descriptive) Statistics

This section focuses on developing methods to include numerical readings, specifically descriptive statistical measures about brushed data items, in an interactive visualization to enhance users’ quantitative understanding of the visualized data. We implemented descriptive statistics overlays in a scatterplot and parallel coordinates view that support analysis and decision making based on important numerical values. The center of data, typically represented by a statistical measure that represents a central or typical value within a dataset, is the most commonly used statistical measure in data analysis. It is easily understandable and valuable for comparisons and decision making. There are several ways the center can be estimated, and depending on the analysis task, different values are appropriate. Examples include the median (the middle number in the set of values), the mean (average), the midrange (the value half-way between the minimum and the maximum), and the mode (the value that occurs most frequently). We compute three different centers: the median, the mean, and the midrange. Additionally, we determine the total spread and the spread based on the standard deviation. Estimating the center and spread, we already have a first useful summarization of the data.

### Descriptive statistics overlays for a scatterplot

An important step is to plan where and how to display the numerical values of the descriptive statistics. Visualization designers should be careful when adding additional

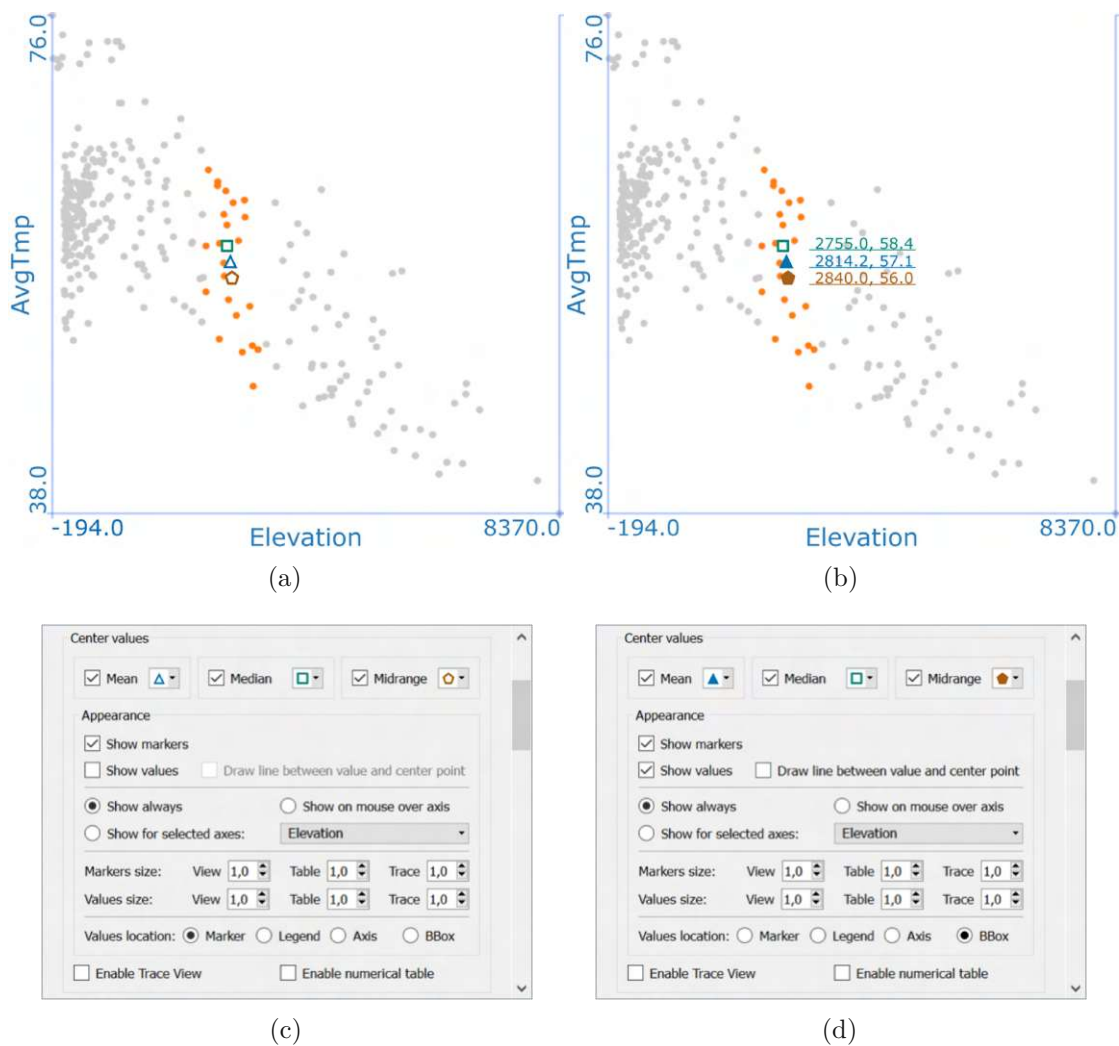


Figure 4.1: Descriptive statistics are shown as overlays on a scatterplot. **(a)** Qualitative readings are enabled by displaying one marker for each of the center values, while in **(b)** precise quantitative readings are also possible through the numerical values shown next to the respective markers. **(c)(d)** A preview of the descriptive statistics setting where the user can configure the appearance of the overlays in the view. The results of the current settings are shown in the respective scatterplots in **(a)(b)**.

graphical elements to existing visualizations to avoid distracting from the already provided and valuable qualitative information in visual analysis. An example of descriptive statistics enabled in a scatterplot is shown in Figure 4.1(a)(b). Three different markers are enabled to differentiate more easily between three different centers of the data. Markers are displayed at the position of the corresponding center value, and we provide numerical values (as labels) next to the marker to enable a quick quantitative reading of each marker. The visualization research community has already proposed many general guidelines for

making a successful data visualization. For example, in their reference visualization model Card et al. [CMS99] state that three elements need to be defined for visual structures: spatial substrate, graphical elements, and graphical properties. This guideline applied to descriptive statistics displayed as an overlay in a view, translates into a two dimensional spatial substrate. Here, graphical elements include text, lines and markers (such as circles and rectangles), while graphical properties encompass the size and position of texts and markers, length of lines, and used colors. We aim to provide users with the freedom to choose how they visualize descriptive statistics.

We implemented a rich palette of options for descriptive statistics overlays. Figure 4.1(c)(d) shows the part of the control menu related to the data center values. Through the provided controls, the user can initiate visualization of not only numerical values but also auxiliary graphical elements. Depending on the task and needs, the user can configure what is displayed in a view. To help the user navigate faster through the control menu, we display a small colored marker next to the name of the associated center value. Using color has several advantages. For example, the numeric values and their corresponding markers are color-linked, meaning they share the same color in the visualization. Color-linking could help establish a visual relationship between the overlaid markers and corresponding numerical values. If the user does not want markers and only numeric values are shown as overlays in the view, he is still aware of the meaning of different values as they differ in color.

Markers used for descriptive statistics can be enabled to help quickly relate and compare between the same or different quantities displayed across several linked views. Markers can be made more prominent by changing their appearance (shape and color). Perceptual discrimination of markers used for the center values becomes better as the markers' size increases. Compare scatterplots (a) and (b) in Figure 4.1 to see how different marker types (filled vs. unfilled) influence how one perceives the distribution of the underlying brushed data items in distinct ways. The area of the shape and color intensity is a fairly powerful visual cue. For a survey of visual queues, including shapes and color intensity, see the work by Cleveland and McGill [CM85]. In the case shown, the objective is to track the changes in data center values while moving the brush. Given that the underlying data items are sparse and displayed with unfilled shapes, the filled markers that represent center values are easier to spot and track.

Koytek et al. [KPV<sup>+</sup>18] utilized lines in a coordinated multiple views system to establish a visual link between data items brushed in a scatterplot (source) and related data items displayed in the linked bar chart view (target). The main goal was to highlight the link between the source and the target through the display of auxiliary lines. In their setup, a bar in the bar chart can correspond to several source data items. A disadvantage of displaying these auxiliary lines is that they lead to unwanted visual clutter depending on the visual layout and the number of overlaid lines. We observed that this technique proves beneficial for specific use cases and particularly when visualizing only a few lines. We adapted their concept to work with quantitative overlays. Our adaptation aims to facilitate and simplify quantitative interpretation by displaying an auxiliary line, which



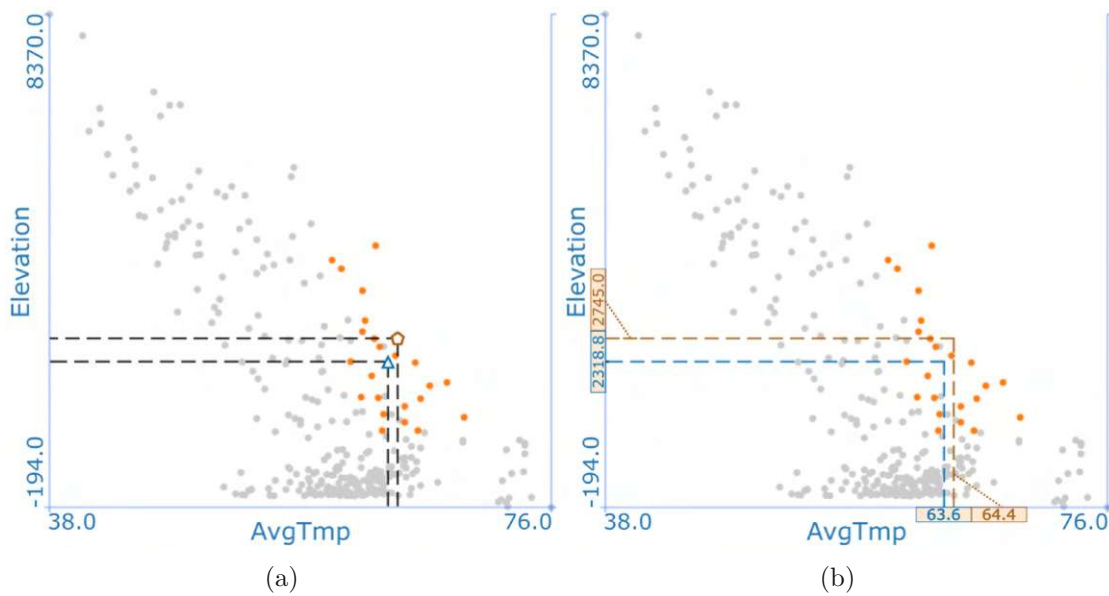


Figure 4.2: Lines are introduced to a scatterplot to facilitate quantitative interpretation of descriptive statistics computed from the brushed data items. **(a)** A line connects the center value of the selected data items, represented by a marker on the scatterplot, to the corresponding position on the data axis (lines are rendered prominently for improved visibility in printed images). **(b)** Labels displayed adjacent to the axes assist users in quantitatively interpreting the two different center values.

aids in precisely identifying the position of the calculated center value within the view and along the numerical axis. Figure 4.2 illustrates two different configurations for displaying auxiliary lines. In configuration (a), lines extend from the positions of the calculated center values to the corresponding positions on the data axes. Since all lines of one data dimension are parallel to each other, visually comparing the difference in values from descriptive statistics within one data dimension is easy. Additionally, we intentionally included markers to emphasize the center value's position within a two-dimensional data space. When the auxiliary lines are enabled, the position of the calculated center can be quickly located by looking at the intersection of the two corresponding lines, as shown in Figure 4.2(b). These lines also intersect with the data axes, allowing users to more easily analyze the center value with respect to the minimum and maximum data axis values. In configuration (b), numerical values for the calculated data center are displayed on the respective data axis. These numerical values can also be shown as overlays on a scatterplot, as previously demonstrated in Figure 4.1(b).

Figure 4.3(b) and (c) illustrate an alternative scatterplot option that we implemented. We show statistical values next to the bounding box (BBox) of the brushed data items, resulting in an appearance similar to that of entries in a typical legend. If this option is activated, the enabled statistics are displayed based on the chosen position—either top, bottom, left, right, or automatically relative to the BBox. The displayed numerical

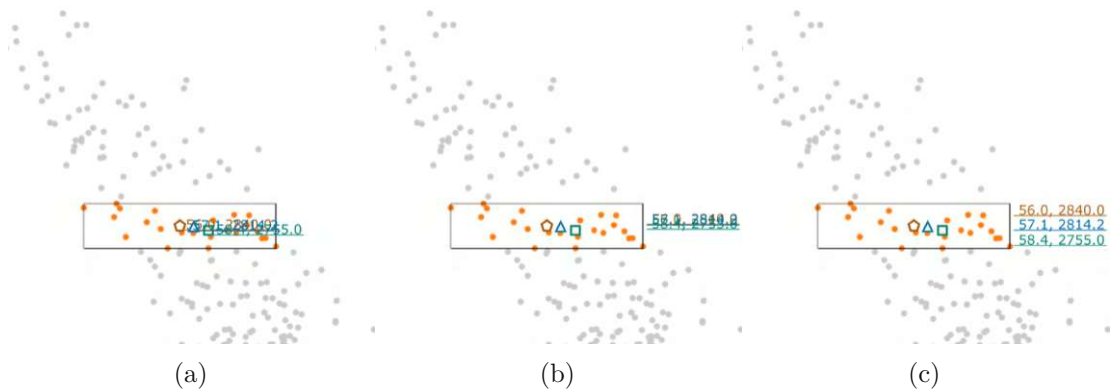


Figure 4.3: **(a)** All three center values of the brushed data are very close to each other and the numerical overlays overlap and are visually unreadable. **(b)** Numerical values are moved away from the markers and shown next to the bounding box of the brushed data. This solves the overlapping of numerical values with the markers. **(c)** In addition, vertical jitter is implemented to reduce occlusion problems for the numerical values. Rendering of the bounding box is optional, i.e., its visualization can be turned off.

values must be positioned adequately, so they do not overlap as in Figure 4.3(a) and (b). We implemented a mechanism for automatically adjusting the vertical position of individual numerical values. As shown in Figure 4.3(c), the introduction of jitter to the vertical coordinates has eliminated the overlap among the displayed values. The jittering technique is applied following the approach used in the beeswarm plot [Ekl]. All numerical values are readable compared to the Figure 4.3(b) where some of the values are unreadable due to overlapping. The additional automation we implemented takes care of positioning the numeric values in relation to the BBox. If the side of the bounding box, where the quantitative values are shown, is close to the edge of the view, the displayed values are automatically placed on the opposite side of the bounding box.

In our implementation, we offer several options for positioning numeric values in a scatterplot: next to marker, next to the bounding box, next to axis, and in the legend. Since brushing is a dynamic process and the distribution of brushed data in linked views will most likely be significantly different between two consecutive brush positions, we recommend displaying numeric values in a fixed place, for example, using a legend. Users can also click-and-drag labels of the center values for more appropriate placement. Interactive positioning is time-consuming and demands the user's attention, making it impractical for utilization during visual data analysis. However, it proved to be a helpful feature when taking screenshots of the current analysis. Although we have not implemented custom legend positioning, it is an option that should be taken into account by visualization designers. Because the legend may obscure the displayed data, it is beneficial if the user can place it at some position where it is least distracting so that he gets a better view of the data. This can be particularly useful in a scatterplot where, depending on the selected data dimension, data items plotted can have very different

distributions across the data space. We also need to take into account potential issues with the occlusion of the numerical values displayed along an axis. In Figure 4.2(b), because the midrange center value (64.4, 2745) is close to the mean center value (63.6, 2318) in both data dimensions, there is not enough space to display both numeric values without overlap at their respective tick position on the axis. Many solutions exist already for adjusting tick labels in graphs. In our implementation, when needed to prevent overlap, a label with the numerical value is automatically shifted vertically or horizontally based on the data dimension. To communicate the shift in the position of a label to the user, we draw an additional dotted line from the respective auxiliary line towards the shifted label. This way, the numerical value is clearly visible to the user.

In our implementation, markers for the data center value have a fixed position on a scatterplot. The position of the marker is determined by the  $(x, y)$  values of the calculated data center, that is, by the respective individual values in the horizontal and vertical dimensions of the data. We use by default unfilled markers for descriptive statistics in a scatterplot because they obscure the data items less than filled markers. ComVis [MFGH08], the framework with which we implemented our extensions, uses a filled circle to display a data item; when a brush selects a data subset, the context is grayed out, and the brushed data items are rendered in orange. The issue with the filled circles and other shapes, in general, is that if a large number of filled shapes is shown in a small area, they can overlap, resulting in what is known as overplotting in a visualization. To deal with overplotting, ComVis provides options to reduce the opacity and spatial extent of a shape used to plot data items. We extend these options to markers as well. For the same reason, many visualization tools, including Tableau [Tab20] and Vega-Lite [SMWH17], use by default an unfilled circle in a scatterplot for the rendering of data items.

### Descriptive statistics overlays for a parallel coordinates plot

Overlays, introduced for a scatterplot, can be easily adapted to work with parallel coordinates or any other view that shows quantitative data. Data axes are presented in parallel to each other within parallel coordinates plot, and data items are displayed along the corresponding data axis in their respective value positions. Consequently, markers representing the calculated statistical values in parallel coordinates are plotted directly on the corresponding axis, as shown in Figure 4.4. The marker and the value for the calculated mean value of the selected data items are shown. As with a scatterplot, the user can combine different settings for descriptive statistics overlays. The options to display numerical values next to marker and next to axis make sense in a scatterplot, but in parallel coordinates, both options produce the same result. Still, the bounding box (BBox) of the brushed data items is a valuable addition in parallel coordinates plot, especially when we do not see hidden outliers due to overplotting. In the example shown, the BBox is enabled for the AvgTemp axis for illustrative purposes, even though it was not necessary since the dataset used is small and we do not have overplotting issues. The option to display numerical values next to the bounding box is also available.

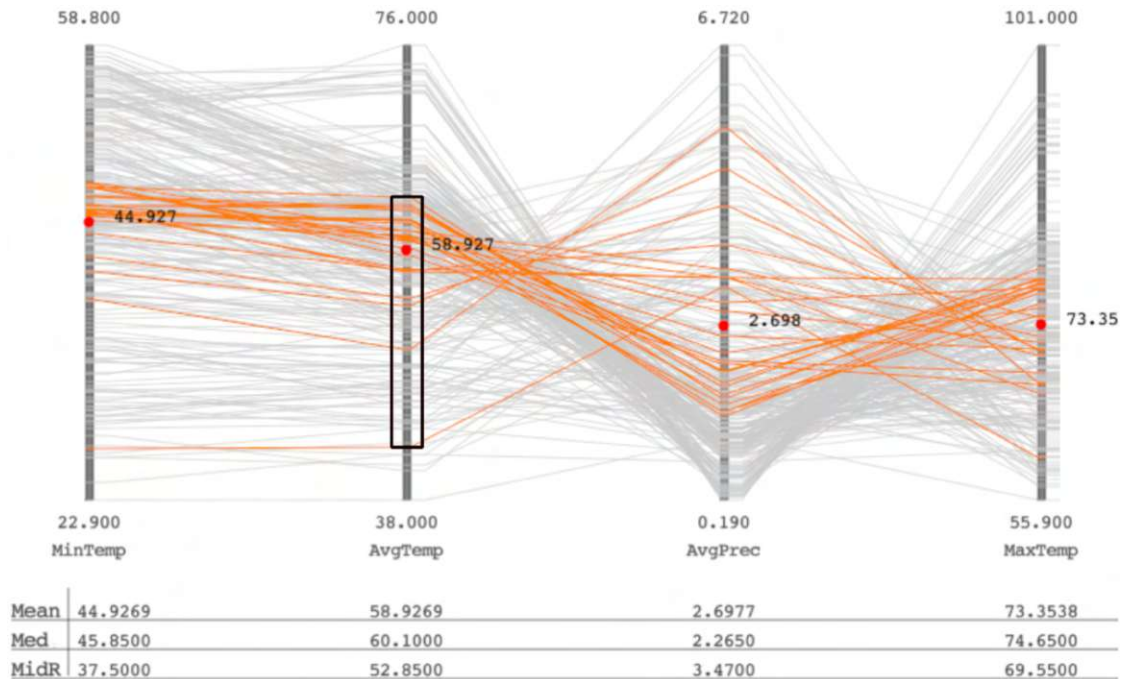


Figure 4.4: The mean value for the brushed data is displayed in parallel coordinates using a red marker to indicate the value’s position on the axis, and next to the marker, a label with the numerical value is shown. The table (legend) below the view displays additional statistics, including the median (Med) and the midrange value (MidR).

Furthermore, we introduced the legend. Within parallel coordinates plot, the legend automatically displays enabled descriptive statistics for all data axes. We do not want to occlude data in the parallel coordinates, so we placed the legend at the bottom of the view, as shown in Figure 4.4. In this way, statistical values for a specific dimension can be easily read.

### Trace View

In the preceding section, we showed several straightforward ways to visually represent numeric values from statistics on currently brushed data. In many cases, the user needs more than just an understanding of the currently selected data to explain a phenomenon. This section explores methods for monitoring trends of a descriptive statistical value.

Some strategies for visualizing sampled data in statistical software avoid directly visualizing all raw data items and instead rely on graphical representations of summary statistics. Choosing brushed data summaries over directly displaying all raw data items allows the selective presentation of the most relevant information to users. This enables them to focus their attention on the most important aspects of the data for the current analysis task. Haslett et al. [HBC<sup>+</sup>91] were among the pioneering authors to utilize this approach

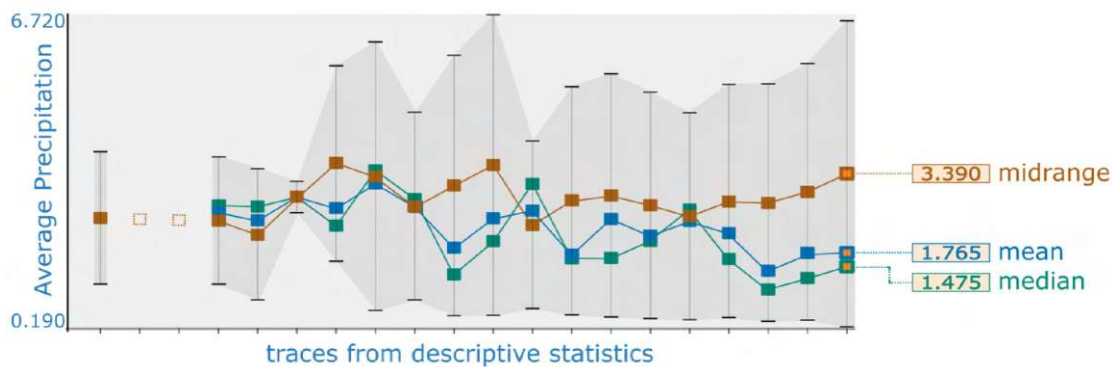


Figure 4.5: The trace view facilitates observation of trends in descriptive statistics measurements derived from brushed data on a chosen data dimension. Active selections within the trace are indicated by an orange-filled marker. Users can choose any marker on the trace for a more in-depth analysis of the recorded values (refer to the text for an explanation). Numerical values are displayed to the right of the graph to enable an exact quantitative comparison.

for supporting interactive visual exploration and analysis with statistics derived from a selected data subset. They employed the moving average technique [HBC<sup>+</sup>91] within a visualization called the *trace view* to depict the trend of statistical values. Specifically, they computed the local average value in the given vicinity of the mouse pointer in the brushed view and represented the result as a *moving average* value added to a curve (refer to as trace) in the trace view. Building upon this methodology, we extended the moving average technique to incorporate the brushed data items in the linked views. Our implementation only considers descriptive statistics derived from numerical data, not from categorical data.

In addition to the moving average proposed in the original paper, we computed three additional estimates for the center of the selected data items, i.e., the median and the midrange, as well as the spread. Consequently, we developed a capability to display multiple traces simultaneously in the trace view, enabling easier visual comparison of the different measurements. Since the trace view offers a historical perspective of changes in the monitored values, we decided to support a quantitative comparison within the trace view too.

Figure 4.5 depicts our trace view, in which each individual trace corresponds to a distinct statistical measurement. Our approach involves calculating a set of descriptive statistics based on the data items associated with the selected data dimension in the linked view while the brush is manipulated in the another view. We display the resulting values as markers and numerical values added to traces in the trace view. The name of the tracked descriptive statistics can be optionally displayed next to the label with the respective numerical value, as shown for three center values (midrange, mean and median). The tracked descriptive statistics share the same name as the corresponding trace, leading to

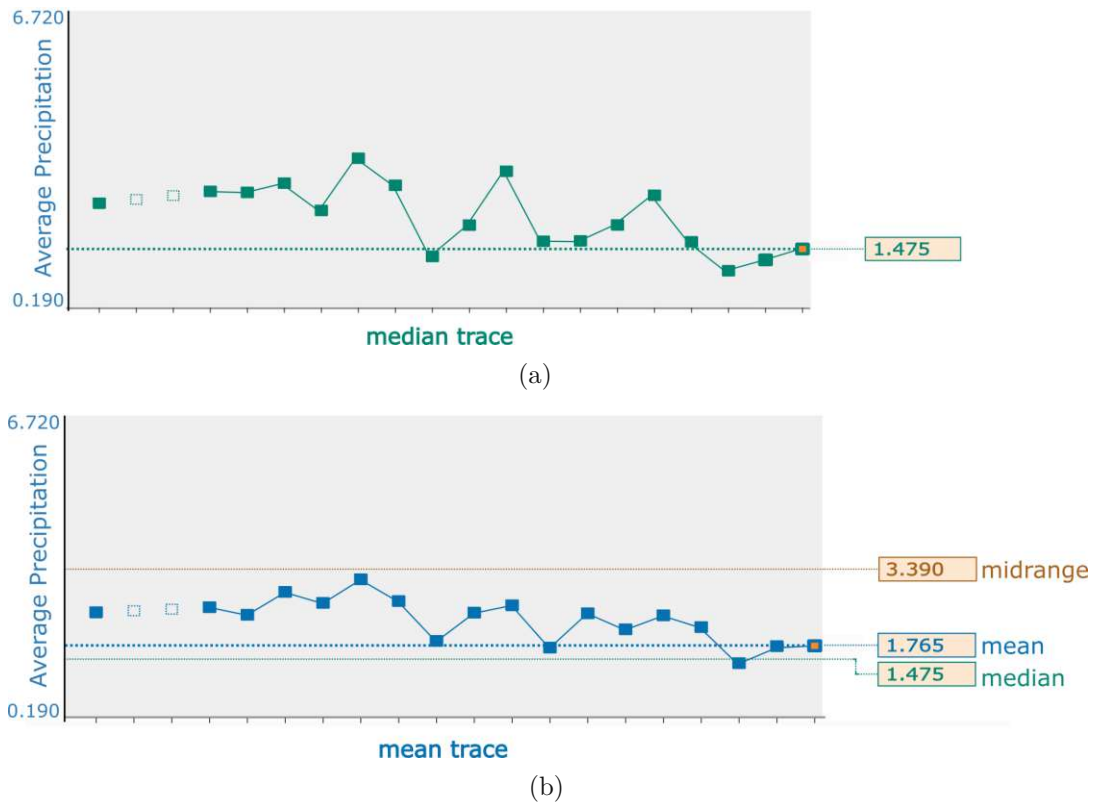


Figure 4.6: Two distinct configurations for the trace view are displayed. **(a)** Trace for the median value is visualized. An auxiliary horizontal line aids in comparing current and previously recorded values on the same trace, providing a clear understanding of differences. **(b)** Besides the tracked mean value for which the historical trend is shown, auxiliary horizontal lines are enabled for two other values (the midrange and median) at the currently active position on the trace.

terms like the mean trace, median trace, and so on. The name of the data dimension from which the data items used to compute the statistical value originate is displayed either next to the vertical axis, as demonstrated in this example, or below the horizontal axis. By default, the vertical axis of the trace view shows the range of data from the dimension used to calculate the statistic. Optionally, the range can correspond to the recorded statistical values on the trace. The trace view can also be used to present the spread of brushed data values used for computing the displayed statistical values. We employ whiskers, a vertical line connecting the pair of whiskers, and a shaded area to illustrate the pattern in the distribution of brushed data values, as shown in Figure 4.5.

Users can customize visual elements in the trace view to suit their specific needs. Figure 4.6(a)(b) shows another example of the trace view's customizability. The trace is displayed only for the key value in the analysis, which is the median center value in (a) and the mean center value in (b). Another visual element incorporated into our



trace view design is an auxiliary horizontal line that runs through the currently selected value on the trace and alongside all other values on the same trace, making it easier to compare current and past recorded measurements, as shown in (a) and (b). A bold, blue horizontal line indicates the mean center value. The brown line, located around the middle of the graph, represents the current midrange value. The green line represents the current median value. These elements are intended to provide users with supplementary information to enhance context and allow for easy comparisons between multiple values. In the shown example, we can quickly conclude that most brushed data items from the current brush have low values for Average Precipitation. This could be easily confirmed by enabling the display of the spread for the brushed data in the trace view, as already demonstrated in Figure 4.5. We chose not to enable auxiliary horizontal lines in the trace view by default to avoid the visual clutter they produce.

When employing a trace view to track multiple statistics, i.e., to show multiple traces, it is essential to have a clear and consistent visual language to avoid confusion and help the user easily associate related information. One effective technique is to use the same color coding for all associated graphic elements, such as markers, values, and labels. To facilitate visual analysis in practice the symbols on the trace correspond to the color selected by the user for the corresponding numerical value. This makes it easier to track multiple statistics within a single trace view and provides a seamless transition when comparing visual information across multiple linked views in a coordinated system.

In the following, we provide an example of how the trace view can enhance the quantitative analysis of brushed data in parallel coordinates. For this use case, we focus on studying average precipitation (AvgPrec) and temperature values (AvgTemp) at different elevations above sea level. During the initial data exploration, we moved the brush freely and slowly up and down along the Elevation axis while observing changes in the two linked data dimensions (AvgPrec, and AvgTemp). We found that data values in the AvgTemp dimension negatively correlate with the elevation values. For most data in the AvgPrec dimension, the same trend was only observable for data values related to high elevation values.

To systematically investigate the identified negative correlation in AvgTemp, we constrained the extent and movement of the brush, providing complete control over the brushing operation. We created a regular grid with 20 cells, i.e., subdivisions on the Elevation axis, as shown in Figure 4.7(a). Instead of displaying traces for three different values in a single trace view, as we did in Figure 4.5, in this case, we configured the trace view displayed in Figure 4.7(a) to show three sub-graphs: mean trace, median trace, and midrange trace. With snap-to-grid on, we moved the brush down from the top grid cell, and accordingly, each brush movement added a new value to the traces in the trace view. Figure 4.7(b) displays the resulting traces. The last marker of each trace is highlighted because it relates to the current brush, which is placed over the bottommost cell of the grid shown on the Elevation axis. The first marker of each trace represents a corresponding center value calculated from the brushed data items in AvgPrec when the brush has been placed on the top grid cell in the Elevation dimension, selecting the

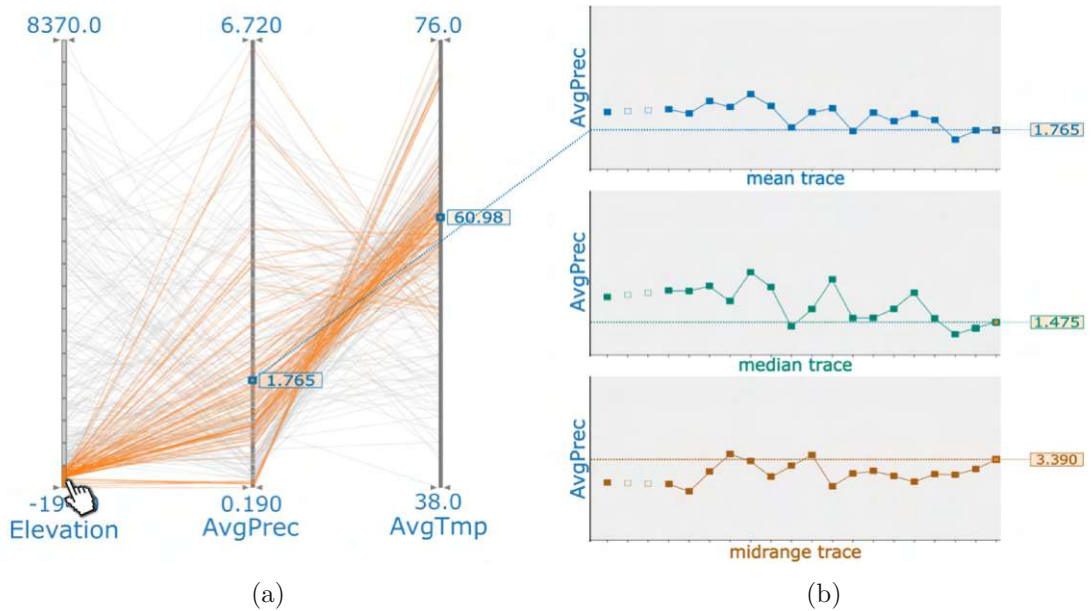


Figure 4.7: Constrained brushing in parallel coordinates aided by traces that support quantitative interpretation of the brushed data. **(a)** The brush advances along the Elevation axis in 20 steps, i.e., distinct elevation ranges from top to bottom. **(b)** The trace view is activated to display the history of changes in summary statistics for the brushed data in the AvgPrec data dimension. For each brush move, summary statistics are calculated from the brushed data and used to update the traces with new values.

highest values within that region. A blue marker with a numerical value in the parallel coordinates indicates the mean value for the selected data along the corresponding data axis, and a line between the mean center value for the AvgPrec dimension and the mean trace in the trace view can be optionally enabled. The line dynamically updates to follow the mean value changes as the brush moves. The trace view was helpful during this initial data exploration because it revealed deviations between values of the computed center values.

We introduced a click-to-brush mechanism, enabling users to select a position on the trace and view the corresponding numerical values or additional information, where available. We also extended the trace view to support the reproducibility of the brushing operation. Figure 4.8 illustrates how the brushing and linking are implemented in the trace view. If a user selects a marker on the trace a linking operation is triggered, resulting in the corresponding data items being highlighted in the parallel coordinates. In addition, associated overlays of descriptive statistics are updated accordingly. By utilizing linking&brushing feature in the trace view, the user can better understand the quantitative meaning of the recorded statistical values and gain insights into the patterns and trends present in the data.

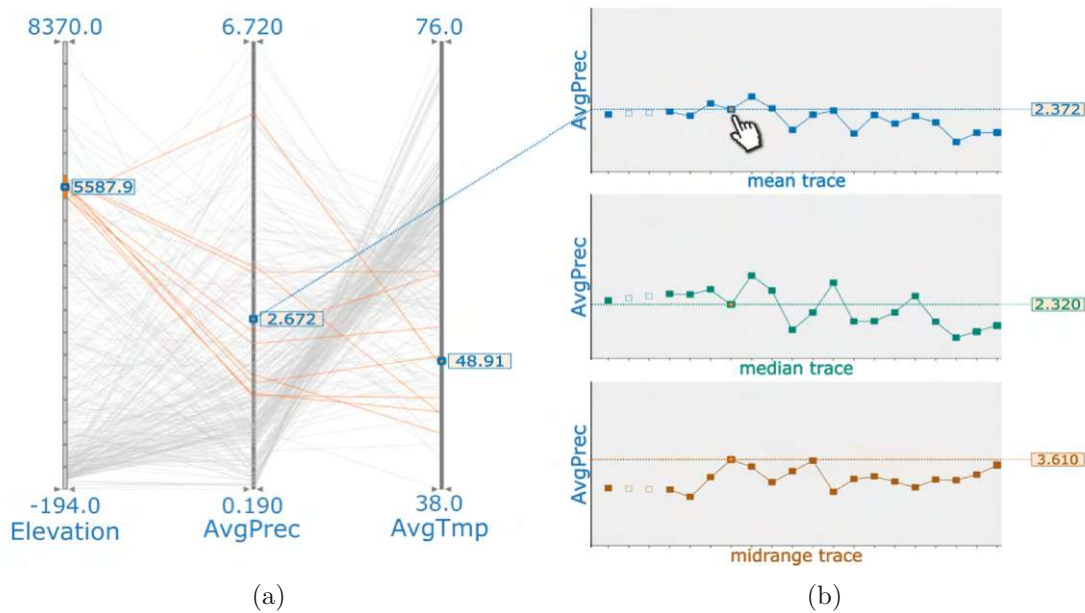


Figure 4.8: The linking and brushing mechanism makes the trace view a potential starting point for provenance analysis. **(b)** In the trace view, each marker represents a descriptive statistical value that was calculated from the brushed data items. **(a)** By selecting a marker on the trace for closer examination, we can easily view and reproduce the brush in the brushed view.

In the following, we describe the elements that make up a trace. Firstly, we have ■ which denotes a value on the trace. For instance, observe the beginning of the trace in Figure 4.9(a). This marker appears either at the start of the trace, i.e., representing the first data value, or after the absence of brushed data items. Secondly, □ represents the absence of brushed data items and indicates that the statistic is not available, as shown in Figure 4.9(b). Thirdly, —■ serves as the second marker for a new value and it indicates the continuation of the brushing operation. Fourthly, — is a long straight-line and denotes that the brush was moved, but the computed statistic remains the same. This situation is highlighted in Figure 4.9(a). Fifthly, —■ is a marker that indicates the currently active brush or a marker selected by the user. For the explanations purpose we used blue symbols in the text.

In the process of implementing the trace view in ComVis, we had to create a mechanism that can distinguish between different types of events and present them clearly and concisely on the trace. This required careful consideration of the visualization and interface technology, as well as the types of events that needed to be displayed. To illustrate important event types, we use data items distributed across three distant regions in the scatterplot, as shown in Figure 4.9. For instance, if the user brushes the top-left cluster as shown in (a) and moves the brush to the top-right region (c), it might

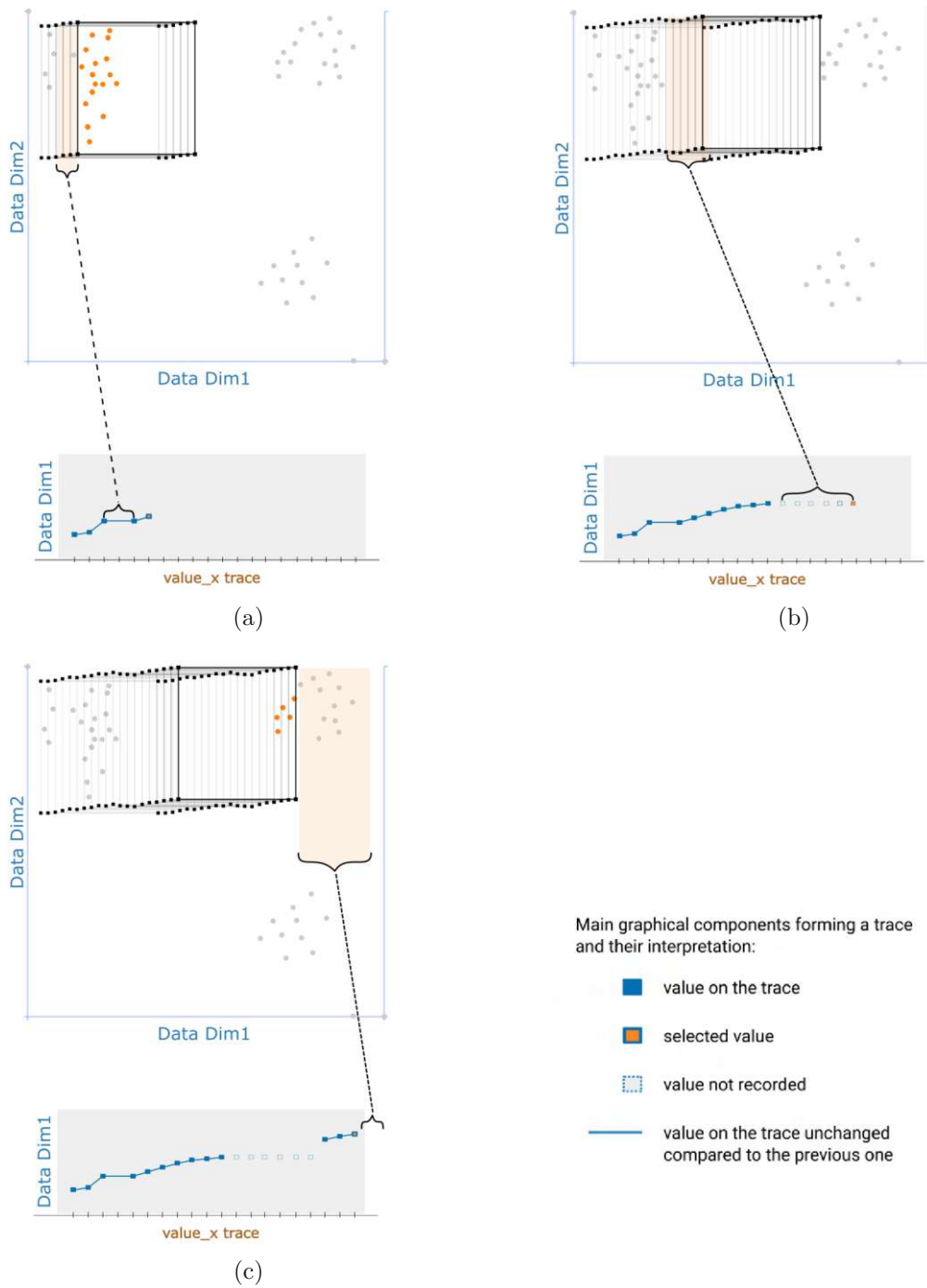


Figure 4.9: An illustrative example of three situations (a)(b)(c) to consider when designing a method for updating a trace in the trace view. **(a)** The subset of brushed data may not change even though the brush position changes. **(b)** The brush is in a position where there is no data. **(c)** The brush continues to move, but the buffer that saves recorded values is now full, or there is no space on the trace to display new values.

pass through the scatterplot's empty middle section (b) which lacks data items. This occurrence corresponds to a brush movement event over empty data space. As a result, it is impossible to compute descriptive statistics from the empty subset of data items beneath the brush rectangle at position (c). The question arises whether to indicate this event type in the trace during brush movement. It is also common that even when the brush is moved to a new location, the current subset of the selected data remains unchanged, meaning that none of the brushed data items exits or enters the shape of the brush. Such events are very common due to the high resolution of modern displays and interfaces, where a significant number of brush movements can be captured before the calculated statistical value changes. This corresponds to an event type where the brush is moved over data items without resulting in an update of the calculated statistical value. Updating the trace view every time the brush is moved can lead to overplotting or displaying irrelevant events for the user. While opting for a slightly reduced sampling is plausible, altering the sampling rate isn't always feasible or straightforward. We considered three methods for updating traces:

1. with every brush move,
2. if the brushed data subset changes, and
3. if the tracked statistical value changes.

In terms of addressing overplotting, continuously updating the trace by free-hand moving the brush is not an ideal solution. Limiting the trace update, i.e., adding a new value to the trace, to changes in the brushed data subset or the tracked statistics can only partially alleviate the overplotting issue. If brushing is used in highly dense data spaces, the trace can still suffer from overplotting despite these measures. To limit the number of updates to the trace, one possible solution is to use structured brushing that constrains the brush's movement to defined steps. This requires implementation and availability of structured brushing in the visualization.

Another alternative that can be applied in the trace view is to limit the number of events displayed on the trace. To achieve this, we have implemented a trace buffer with customizable size. This allows the user to utilize all three mentioned methods for updating traces without caring about overplotting.

The distance between all displayed values on the trace is the same, and is calculated by dividing the available space on the horizontal axis into divisions of equal size, with the number of divisions corresponding to the size of the buffer. We ensure that when displaying multiple traces in the same plot to compare different values, they have the same length and equally spaced ticks and labels. The buffer serves as a container for storing the definition of the visual elements that comprise the trace, corresponding numerical values, and the information about the brush used in producing the value. We use the information about the brush so that, at the request of the user, i.e., when brushing a trace, the brush can be reproduced in its original view. The user can influence the

total number of values shown on the trace by altering the buffer size. For example, the buffer size 100 means that up to 100 values can be added, i.e., displayed, on the trace. Additionally, the user can clear the trace view, i.e., remove traces, by emptying the trace buffer, which can be particularly useful when beginning a new analysis task or requiring a rerun. If the buffer is fully filled, the oldest value is deleted, and the most recent value is appended, similar to the functionality of a process monitoring dashboard in Microsoft Windows.

The trace view can be integrated as a standalone view, included in a separate view container in a coordinated multiple views framework, or as an extension to existing views. By juxtaposing the trace view within the view container of an existing visualization, users can quickly display relevant information on demand. This layout also reduces cognitive load by presenting additional information from the trace view where needed, i.e., adjacent to the data dimension's visualization for which the summary statistics are enabled. It is possible to enable multiple trace views, one for each data dimension that requires analysis. For instance, if using a scatterplot in a linked view, the user can enable traces for the horizontal and/or vertical data dimension, or select from one of the visualized data axes in parallel coordinates.

Alternatively, incorporating the trace view as a standalone view enables two key aspects. Firstly, it allows multiple independent trace views to display trends of summary statistics for brushed data across diverse data dimensions. This means that we can display any data dimension in the trace view, regardless of whether a data dimension is selected in a linked view or not. Secondly, it provides the flexibility to organize trace view containers in user defined arrangements. Both layouts aim to improve linked visualizations by providing additional quantitative and statistical information.

### Scatterplots with Descriptive-Statistics Trace

The trace view discussed in Section 4.2 can be thought of as an independent supporting graph that provides quantitative information for any linked data visualization. Here, we propose a customized approach that involves displaying traces from the trace view directly in a scatterplot, aiming to enhance the functionality of a standard scatterplot.

Descriptive-statistics trace is a two-dimensional visual representation of the history of computed statistics from the brushed data overlaid directly onto the scatterplot. Because the traces are displayed over the brushed data, we assist the user in maintaining focus on two elements simultaneously: the brushed data and the descriptive statistics that characterize it. It is essential to ensure that descriptive statistics added to interactive visualizations used in visual analysis can be understood by users without substantial statistical expertise. This can be supported through careful visualization design.

The key objective of the trace from descriptive statistics is to offer the user an enhanced understanding of changes in the brushed subset of the data displayed on the scatterplot while the brush is moved to different positions in the brushed view. By observing the trace,



and the additional information we provide along the trace, users can gain qualitative and quantitative feedback and improve their interpretation of the brushed data and hidden correlations.

To further enhance the effectiveness of the proposed approach, we draw upon the techniques previously discussed in Section 4.2, as well as the concepts introduced in the original trace views paper [HBC<sup>+</sup>91] and the scatterplot with error bars approach [Hin07]. For example, in statistical applications, it is common to display summary statistics as added values with error-bar lines rather than all data items. Such approaches use, for example, a clustering variable to divide the data items into groups for which summary statistics can be generated. Error bars are often mentioned and used in histograms. Various error bars have already been introduced to support different statistics concerning data items. An example is emphasizing the variability of both y-values and x-values for each item, or displaying related quantities such as confidence intervals, standard errors, and standard deviations. Error bars in our work represent the variability associated with each data center value, in concrete, the standard deviation of the data, the standard error of the center value calculation, or the data range. The implementation can be easily extended to some other statistical value available in a scatterplot.

The advantage of implementing a trace within a scatterplot is that it enables the display of descriptive statistics from both visualized data dimensions using a single trace. A marker on the trace represents a two-dimensional value. Unlike the trace view, which focuses on a single data dimension, calculating descriptive statistics for the scatterplot's trace involves data items from both displayed data dimensions. Figure 4.10(a) illustrates an example of implemented descriptive-statistics trace in a scatterplot. The brush was moved along a straight line through the scatterplot in the brushed view (not shown here) and positioned in 6 different locations. The blue trace was generated by connecting the mean values calculated from each brushed subset in the shown scatterplot as the brush was moved.

We also introduce the concept of the reference trace. It can be used to quickly visually check the deviation of statistical values obtained from the data relative to the same data's reference values. We may not have reference values in practice, so we have to define them somehow. One way for the system to obtain reference values is to allow users to input values into the system. As an example to illustrate the concept, we will use the linear reference trace, which we can easily calculate programmatically. Figure 4.10(b) shows the reference trace. The reference trace was created by linear interpolation between the mean values in the initial and final brush positions in the linked scatterplot. We do that to illustrate how linear movement in the brushed view does not necessarily produce a linear trace in the linked view. There are some differences, although not huge, between the reference trace and the blue trace.

The user can depict paths in different colors to enhance visual distinguishability. If the user wants, he can analyze only the paths from the calculated statistics by altering the opacity of the plotted data items to hide them in a scatterplot.

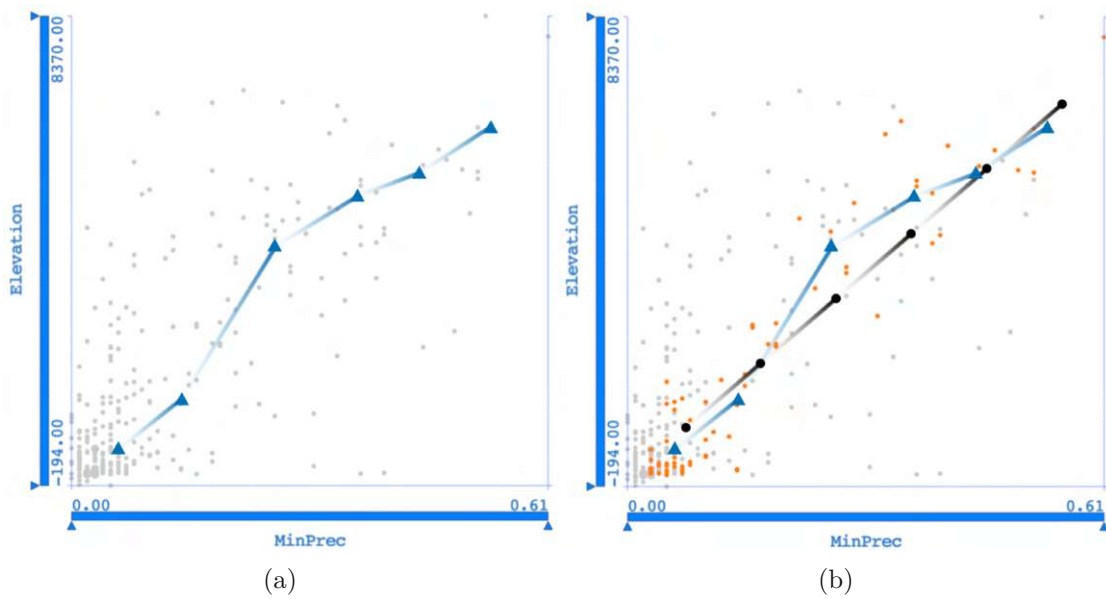


Figure 4.10: Placing the trace as an additional layer on top of the data visualized in the scatterplot allows the user to observe how the value of the descriptive statistics progresses in the context of the two visualized data dimensions. **(a)** Trace of the mean value. The moving mean value is represented with a blue triangle. The direction of the trace in the visualization is indicated using a gradient. **(b)** The reference trace is shown in black and the brushed data items along the trace are displayed in orange.

We encoded the direction in the trace’s visualization. This is particularly valuable for users as it helps them quickly identify the next value on the trace. To encode the direction of the trace in the visualization, we used a gradient which starts off colorless and saturation gradually increases as it approaches the next value on the trace, allowing for easy recognition of the trace’s direction.

Changes in the linked views can be significant between consecutive brush positions in the brush view. The trace depicting the mean values in Figure 4.11 shows sizeable differences between several positions, as well as in comparison to the reference trace. Such a change causes sudden jumps in the linked view, and distracts the user. This distraction exacerbates the mental image creation. We use the cross-hair to enhance perception and understanding of changes in the linked view. In combination with the animated brush, we propose to animate the cross-hair transition in the linked view in order to prevent a distraction of the user.

The cross-hair consists of thin vertical and horizontal lines positioned at the observed value on the trace. It can display either the standard deviation from the observed value or the range of brushed items when positioned in the middle of the subset. In our implementation, we allow only one cross-hair in the view at a time to avoid visual clutter. The exception is when the cross-hair is animated to enhance perception and

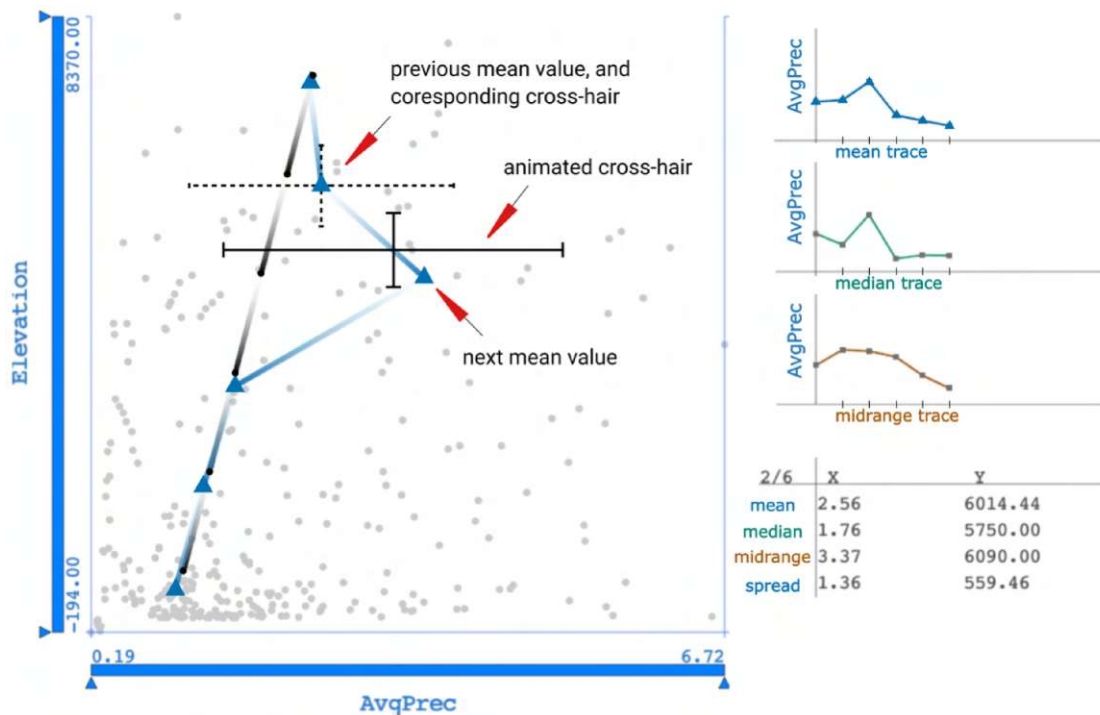


Figure 4.11: An example of analyzing changes in statistical values within a linked scatterplot using animated transitions. The cross-hair moves smoothly along the trace. The animation helps in perceiving the transition. The numeric values in the table change dynamically based on the position of the cross-hair.

understanding of a value change along the trace, as the following example explains. In Figure 4.11, two cross hairs are displayed. The cross-hair stays at a brush position for some predefined time, then it smoothly moves to a new position. The cross hair with dashed lines is optional and represents the previous brush's data items. It disappears as soon as the animated cross-hair reaches a new position on the trace. The cross-hair with thick black lines is for the brush's data items of the current animation frame. While visualizing the transition, the cross-hair adjusts its size through interpolation and changes its direction based on the next value along the trace. Depending on the user preferences, the path is drawn as the animation evolves, or the complete path is shown and the cross-hair moves along the path, as shown in the current example. This visualization of the transition does not only help in eliminating distraction, it also actively amplifies cognition of the trend evolution. A case study is needed to quantify the impact.

In order to quantify changes of data center values and spreads with respect to a moving brush, we present the numerical values for the current brush in a tabular format. Subsequently, as the brush moves, both the summary of descriptive statistics and the table containing the numerical values automatically update simultaneously.

### 4.3 Relative Difference Plot

Analysts are often interested in quickly contrasting the current and the previously recorded value, or some other value the analysts want to compare against. Earlier in Section 4.2, we discussed the visualization of descriptive statistics calculated from the brushed data items. We displayed these numerical values directly in the linked views as overlays and in the trace view. They enable monitoring changes in the selected statistical values through time. If there are changes in value, the difference can be expressed in absolute or relative terms. In this section, we concentrate on depicting relative changes rather than absolute changes discussed in the previous sections.

Numerous examples in the visualization show the expressiveness of derived relative values. An example on trade balance by Tamara Munzner [Mun14], demonstrates how visualizing the difference between exports and imports could be much more effective for the task of trend analysis as the representation of both absolute values side by side. To facilitate visual trend analysis, she first derived trade balance values from the original data, i.e., from export and import values, and then visualized the derived relative values in a separate plot. In other words, instead of plotting two curves for the original data and visually comparing their differences, she provided a single curve view showing relative values that the user can easily understand.

We now introduce the relative difference plot. It shows changes in observed quantitative values in relative terms. The user can quickly see if the value is close to the reference or not. For simplicity and to demonstrate our approach, we concentrate on analyzing the descriptive statistics from the brushed data. We are interested in observing changes in the values of the data center, and spread along the path of the animated brush created in a scatterplot. We decided to implement the relative difference plot as a two-dimensional plot that requires at least two quantitative values, including one original and one reference value. The standard formula for calculating the relative change takes a previously calculated value as a reference, against which the current value is then compared, as shown in Equation 4.1.

$$\text{relative change} = \frac{\text{current value} - \text{reference value}}{\text{reference value}} \times 100\% \quad (4.1)$$

In Equation 4.1, the reference value is used as the denominator. However, a concern arises when the reference value is zero. In such cases, using zero as the denominator could lead to mathematical issues, such as division by zero. Alternative approaches or adjustments may be needed to handle situations where the reference value is zero. Depending on the use case, current value and reference value can be chosen based on existing data items or newly derived data. In this work, a reference value is some value that we expect to find in the currently brushed data items. For simplicity, we compute the reference value by linear interpolation between the values of descriptive statistics calculated in the initial and final brush positions in the linked scatterplot.

The fundamental concept of the relative difference plot is depicted schematically in Figure 4.12, where three different plots are shown. (a) is the brushed view, in (b) is the

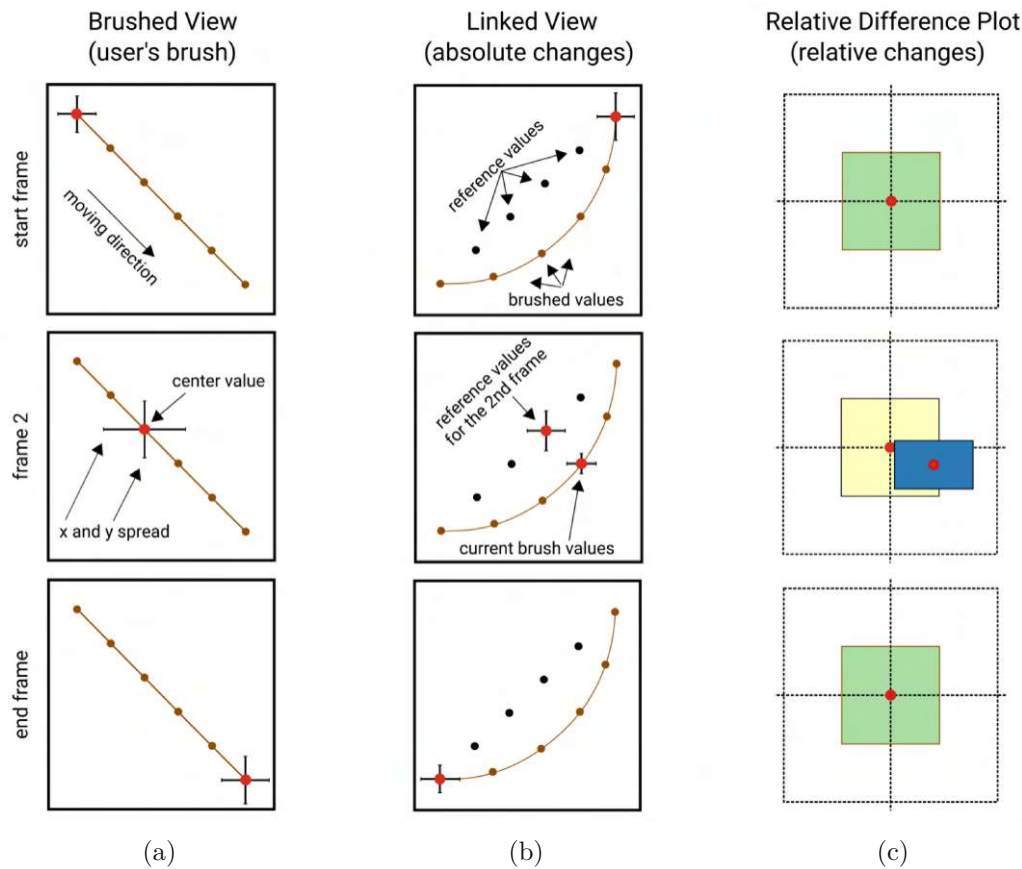


Figure 4.12: While the brush moves in the brushed view (a), the user observes absolute changes in the quantitative measurements in the linked view (b). The center point is marked with a red dot and the spread with a black cross-hair. The difference between the current and reference values can be determined visually by contrasting the two cross hairs and positions of the center values. In the relative difference plot (c), we use a filled rectangle to emphasize spread, but the cross-hair is also possible. The blue rectangle represents the current spread, yellow represents the reference, and green indicates when the compared values are identical, as seen in the first and last frames of the animation. Instead of an absolute difference, the change in value is relative. Details are described in the text.

linked view, while the relative difference plot is in (c). The arrangement of the views is arbitrary. They are connected via the linking and brushing mechanism. Any brush movement in the brushed view will update the visualization in the linked view and the relative difference plot. As shown in Figure 4.12(a), the animated brush moves along a linear path in the brushed view. To better convey this dynamic change to the reader, we have shown three different positions of the brush: the initial position in the top-row, the last position in the bottom-row, and one intermediate position, denoted by *frame 2*.

Instead of showing the reader the brushed data items at all positions of the animated brush, we have simplified the visualization and only display the computed data center value, and the spread. We interpolate center positions and horizontal and vertical spread values to create a set of reference values, for example, as shown in the middle-row in Figure 4.12(b). Brushed data in the linked view are displayed along a brown curve, emphasizing the differences from the interpolated reference values. As the animated brush moves through the brushed view, we compute the linked data center value and spread values and depict them in the relative difference plot for each brush position.

The analysis task involves comparing the data center value and the spread of brushed data for each animation frame with the corresponding reference values. In (b), visual comparison of the two cross hairs and the positions of the center values allows for the identification of the difference between the current and reference values. In the relative difference plot, comparing current values with the reference values is easier for the user, as only the visual representation of the current values changes. For example, the reference rectangle is consistently displayed using a unit scale and placed in the center of the view.

The reference values at start and end frame match the current values. This depends on the method used for calculating the reference values, in our case, as noted above, it is a linear interpolation. Alternative methods for providing reference values can also be used, such as comparing the value at the current brush position with the value at the previous brush position.



# Demonstration

To evaluate the effectiveness of the proposed techniques for analysis tasks requiring reproducible results and quantitative readings, we conducted two demonstrations at different stages of the study. The first demonstration was performed at an early stage of implementation to provide insights and guide further research. The second demonstration was carried out at the end of the study, using the complete set of techniques available in the ComVis [MFGH08] visual analysis tool. The results of these two demonstrations are presented in the following two sections. The first Section 5.1 focuses on climate data, while the second Section 5.2 examines country indicators. In the practical application of the methods discussed in this work for visual analysis, we applied them in the context of mechanical power transmission in cars. The results of this application are presented in a separate research paper [RSM<sup>+</sup>16]. A brief summary of the published results is provided in Section 5.3.

## 5.1 Initial Use Case: Climate Data

The goal of the preliminary demonstration was to collect user feedback in the early stage of development, identify areas for possible improvement, and define the necessary research steps that will bring us closer to reproducible and quantitative visual analysis results. Once the test environment was established, we invited a visualization expert with extensive knowledge of the climate data. The expert evaluated the new techniques as part of an investigation of multi-run climate data generated at the Potsdam Institute for Climate Impact Research.

The data we used was acquired to study the potential climate change caused by a meltwater outbreak event from proglacial Lake Agassiz. This event passed through the Hudson Strait about 8260 years ago, resulting in a cooling of approximately 3.6K over the North Atlantic. Climate changes occur over an extended period of time, ranging from decades to millions of years, making a combination of data analysis, computer simulation, and models essential

for their study. A climate simulation that attempts to explain this interesting event is the PIK Climber 2.3 model. It is a coupled atmosphere-ocean-biosphere model of intermediate complexity that provides aggregates for 35 important climate data over the time period of 500 years, including  $CO_2$  concentration, global surface air temperature, Greenland temperature, and global precipitation. Two diffusivity parameters, both pertaining to the oceanic component of this model, are utilized as independent parameters that influence the climate simulation: one horizontal parameter referred to as  $DiffuH$ , and one vertical parameter referred to as  $DiffuV$ . To better understand how these parameters influenced the simulation, each parameter was varied ten times, and a total of 100 ( $10 \times 10$ ) simulation runs were computed. Various representations, such as histograms, boxplots, and scatterplots, were used during the analysis. In the preliminary demonstration, the new techniques were implemented only for the scatterplots, about which we informed the invited analyst. The visualizations were configured so that a scatterplot shows the independent parameters ( $DiffuH$  and  $DiffuV$ ) while the other views display a selection of the dependent parameters. All views were linked to facilitate coordinated analysis.

Figure 5.1 shows four scatterplots, each depicting a minimum temperature aggregate on the horizontal axis and a maximum temperature aggregate on the vertical axis. The temperatures shown are ocean surface air temperature (OceanSurfAirT), Greenland temperature (GreenlTemp), tropical temperature (TropicT), and Antarctic temperature (AntarctT). Each data item, i.e., point in a scatterplot, represents minimum and maximum values with respect to one simulation run. The distributions of these temperatures are substantially different. The ocean and tropical temperatures show an expected correlation between the minimum and the maximum values. The situation is more complex for Greenland. The data consist of two distinct clusters of values. Each cluster represents either low or high values. Interestingly, both clusters encompass the entire range from minimum to maximum values, i.e., for both, low and high, minimum and maximum values, the respective other categories exist. There is no clear correlation. The correlation between the minimum and maximum values for Antarctic temperatures is highly distorted for higher minimum values.

The analyst conducted an interactive analysis by applying brushing to the highest quintile of tropical minimum temperatures. He used a quintile grid and snap-to-grid brushing, as illustrated in Figure 5.2(a). Additionally, the independent variables were analyzed, see Figure 5.2(b), as well as the box plots of mean values of different temperatures, see Figure 5.2(c). It was quickly discovered that high  $DiffuH$  and low  $DiffuV$  values are the primary factors contributing to the observed distinction between maximum and minimum temperatures in Greenland. Additionally, intriguing anomalies were observed in the box plots (Figure 5.2(c)). All mean temperatures except for Greenland and Ocean Volume (VolAvgOceanT) raised in this case. Figures 5.2(d-g) show the four temperature scatterplots with overlaid descriptive statistics. They correspond to the highest quintile of tropical minimum temperatures. They reveal some correlations of tropical and ocean temperatures, but a less clear picture with respect to the Greenland and Antarctic temperatures.

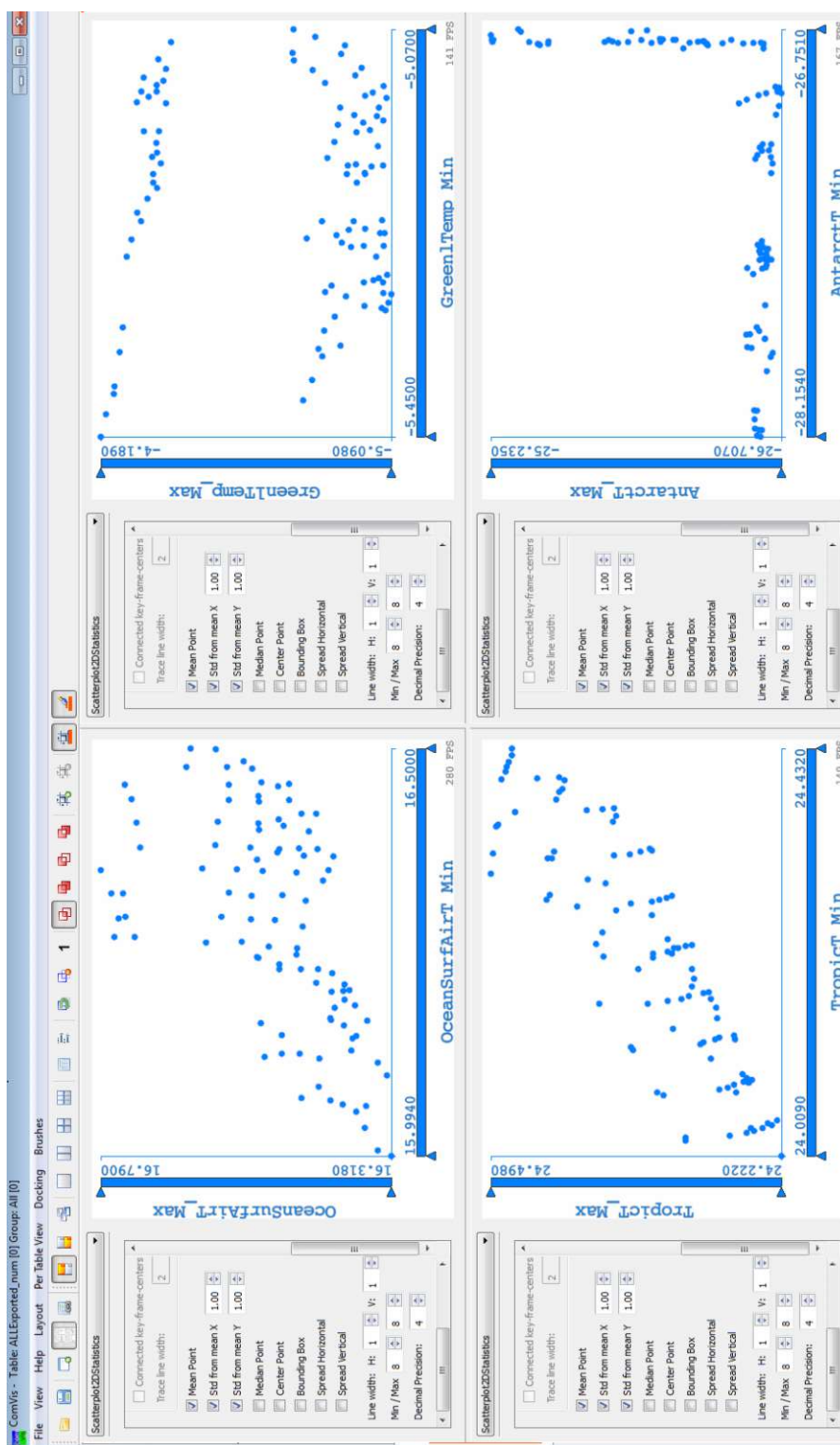


Figure 5.1: Climate temperature data from the four observed geographical regions are displayed for comparison using four scatterplots.

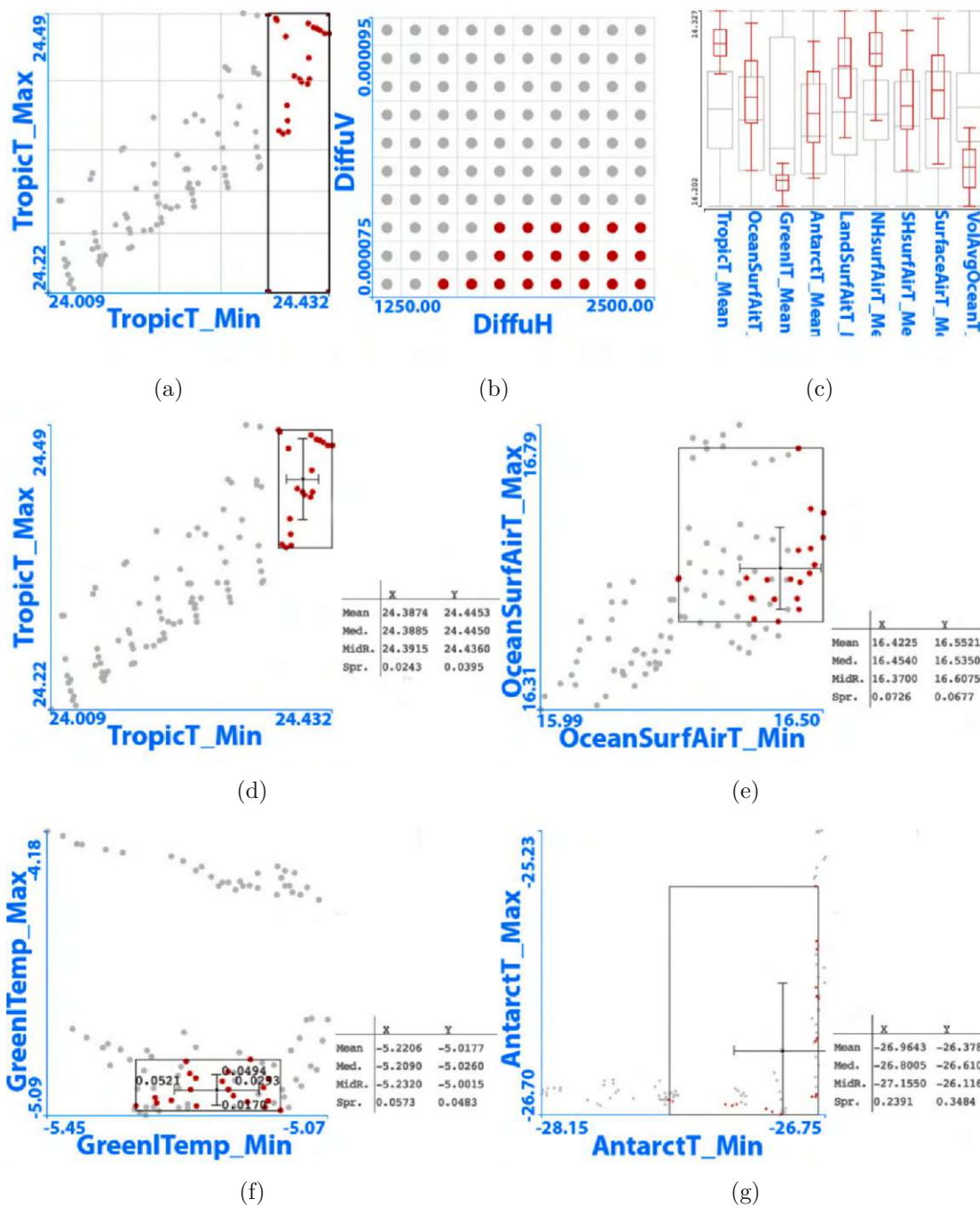


Figure 5.2: Climate anomaly analysis. (a) Scatterplot shows a quintile grid and the snap-to-grid brushing option enabled, allowing selection of all data items in the highest quintile of minimum tropical temperatures. (b-g) Overlaid descriptive statistics are also enabled in the linked scatterplots to display critical characteristics for the brushed data items related to four distinct temperature measurements.

As part of the analysis, all remaining quintiles were examined by brushing within Figure 5.3(a). The most interesting correlations were found in the lowest quintile of maximum tropical temperatures. Box plots in Figure 5.3(c) and independent variables in Figure 5.3(b) now reveal a different picture. Figures 5.3(d-g) display temperature scatterplots for this selection. Once again, a certain degree of correlation can be observed between ocean temperatures and tropical temperatures. A more complex relationship with the temperatures of Greenland and Antarctica was confirmed.

The next task was to examine the influence of diffusivity parameters on tropical temperature changes. In this context, the analyst felt it necessary to examine the data along the distribution contours depicted in Figure 5.3(d). The Mahalanobis brush proved valuable for visually identifying and analyzing trends within this specific range of data. The analyst used the Mahalanobis brush to encircle the tropical values range, i.e., to make one round around the items on a scatterplot representing data related to the tropical temperatures, consistently encompassing 20% of all data items in the brush. Such a round would be very complicated to do using conventional brushing only. Here, it was a success right on the first attempt as the Mahalanobis brush effectively aided in isolating distinct temperature value ranges where the values of the independent parameters diverged. Figures 5.4(a-g) illustrate the utilization of the Mahalanobis brush for analyzing tropical temperature values, along with the corresponding visual updates in the linked scatterplots displaying data for the two independent parameters. The linked data splits into two clusters upon reaching higher maximum temperature values. There is a clear difference observed in terms of values between lower and higher maximum values, as seen in Figures 5.4 (b) compared to (f), (c) compared to (e), and (a) compared to (g). Other linked scatterplots not presented here, which display additional data dimensions, reveal that brushing through higher tropical maximum temperatures results in a broader spread across several dimensions.

During the demonstration, reproducibility of brushing operations was considered the most important need. Since we had only one expert available for evaluation, we also aimed to further investigate the reproducibility of the brushing results created with the newly developed techniques. During the analysis process, we created several screenshots showing the results of the expert analysis. As part of this investigation, we attempted to reproduce the previous analysis results by viewing the stored images. With little or no effort, we managed to obtain the same results. The analyst's feedback was valuable in identifying the strengths and limitations of our brushing techniques. Specifically, with the Mahalanobis brush, he noticed that even small deviations in the cursor position can cause variations in the selected data items. As a result, the only way to adjust the selection from the Mahalanobis brush was to slightly move the mouse and realign with the previous brush position. At that point, the only available user interface control was the percentage parameter, which determines the amount of data selected by the brush from the total dataset relative to the cursor position. To address this limitation and enhance accuracy, we decided to expand the interaction options and give users more control over the Mahalanobis brush. In addition to the percentage parameter, we



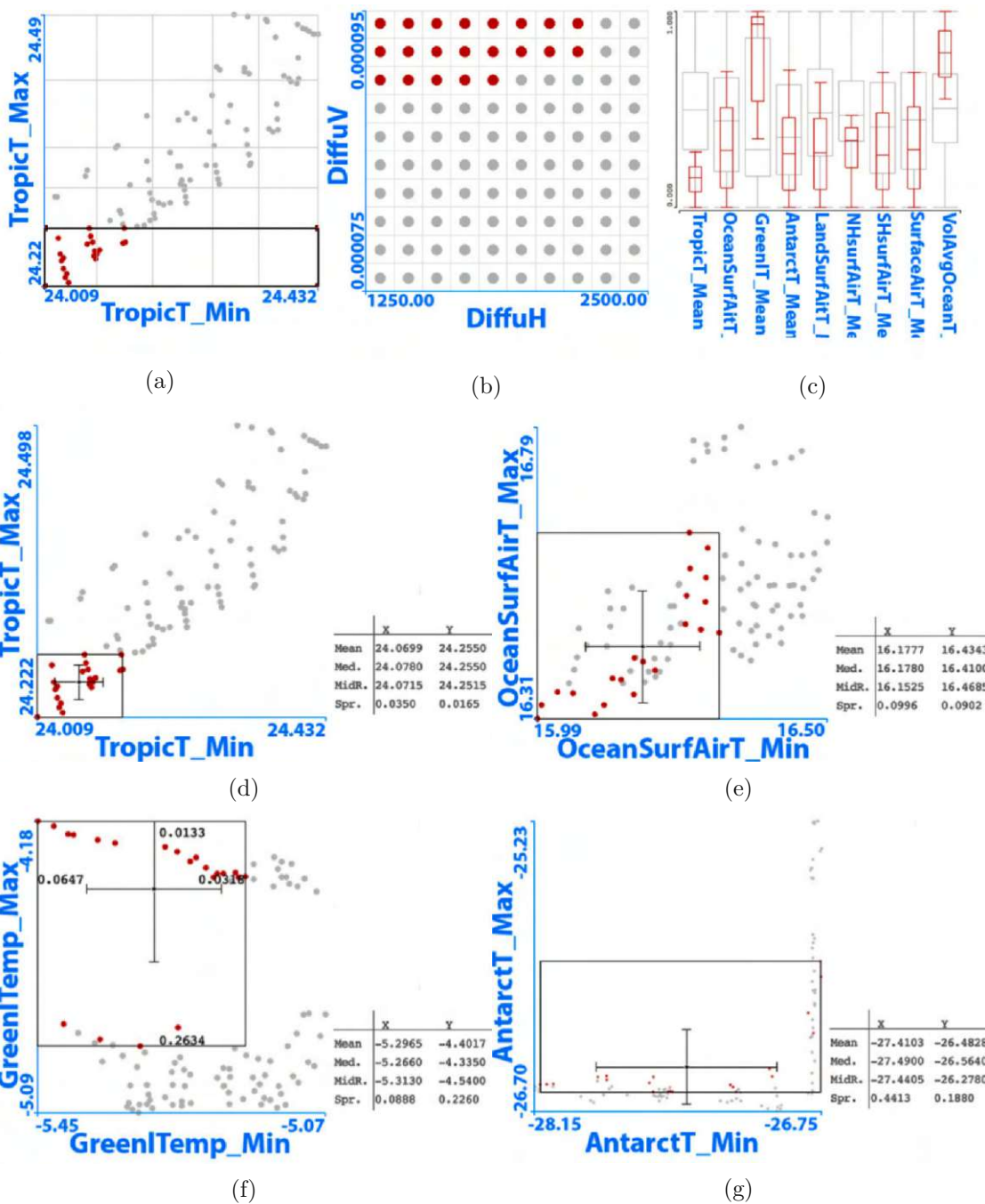


Figure 5.3: (a) A quintile grid and the snap-to-grid brushing option are enabled on the scatterplot to select all data items within the lowest quintile of maximum tropical temperatures. (b-g) Professional analysts base the analysis on interesting subgroups. Providing these analysts with user-friendly tools for effortless subgroup selection and for generating results enriched with descriptive statistics significantly enhances the potential for analysis.



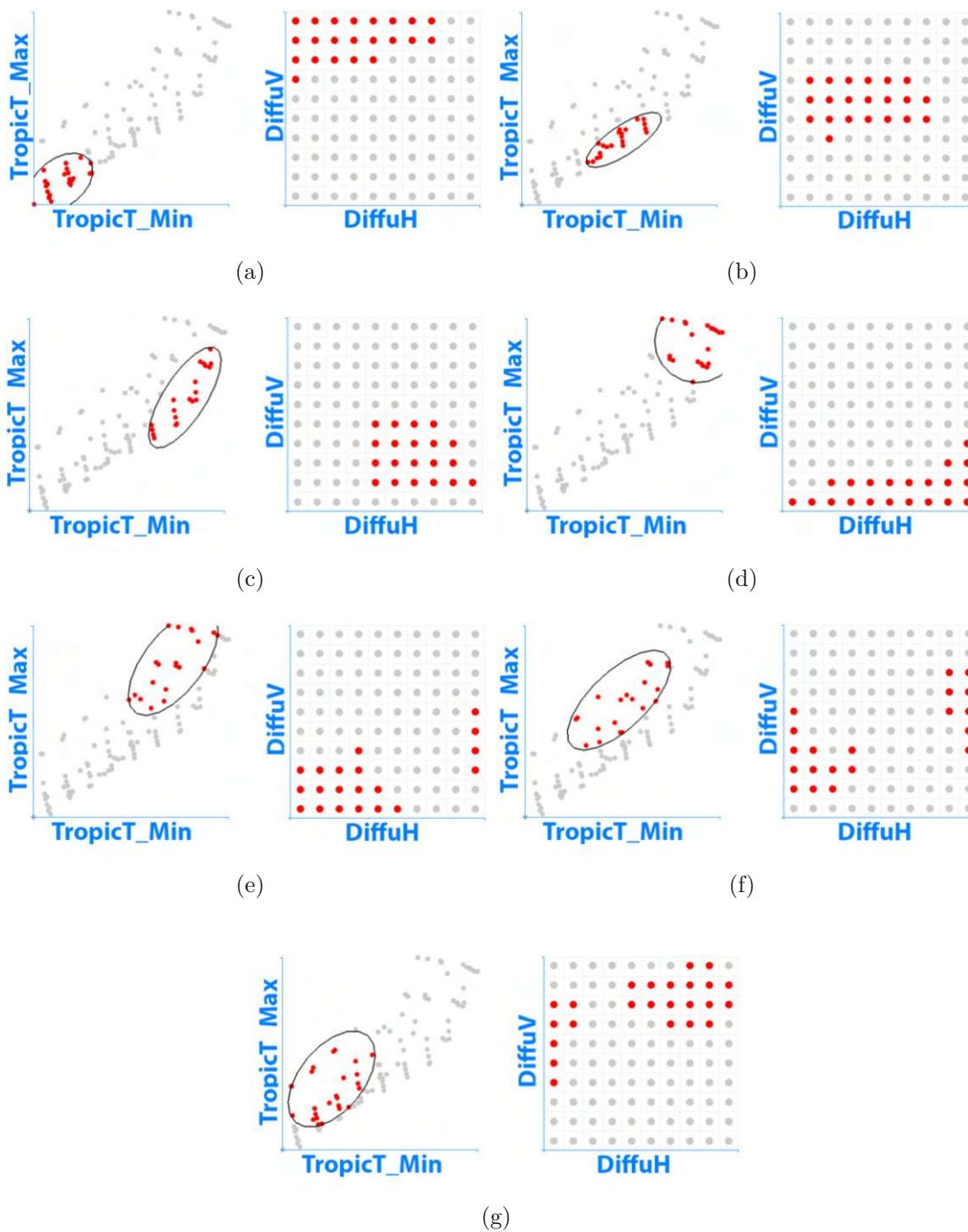


Figure 5.4: **(a-g)** Utilizing the Mahalanobis brush to encircle the range of tropical temperatures, consistently encompassing 20% of all data values. **(e)(f)(g)** The *DiffuH* and *DiffuV* values split into distinct groups when (relatively) higher max-temperatures are brushed.

introduced the sensitivity parameter, which can help mitigate the impact of small changes in cursor position. With this parameter, users can specify how many data items from the local distribution the Mahalanobis brush will consider for orientation calculation. Using a smaller value increases the sensitivity of the Mahalanobis brush to changes in the immediate vicinity of the cursor position.

The analyst took advantage of the constrained brushing space available in the scatterplot. He provided feedback indicating that constrained brushing makes it easier to select quintiles or other common statistical subgroups, resulting in a more efficient analysis. Furthermore, he commented percentile brushing as a significant contribution to visual analysis, as it enabled the constant selection of the same number of brushed data items. Another comment was related to the use of constrained brushing techniques. These techniques enable faster and better reproducible analysis, facilitates the understanding of results, and supports decision making even without the support of the overlaid numerical values. The analyst asked for these techniques to be extended to other types of visualizations beyond scatterplots.

Regarding the snap-to-grid feature, we found that expanding the possibilities of supported brushes beyond the rectangular brush could be very important for a wider acceptance of this technique. Additionally, it will open up additional brushing possibilities, such as positioning the circular brush in the scatterplot at the vertices of the grid. Encouraged by the positive and constructive early feedback we received during the preliminary demonstration, we continued to refine our techniques to ensure optimal performance and usability. Our objective was to enhance their utility for users of IVA.

## 5.2 Use Case: Exploring Statistics Data of Countries

After implementing the comments from the preliminary feedback and adding new details that we thought could improve and complete the contribution of this work, we prepared a new demonstration. This time we invited four participants with a solid knowledge of visual data analysis. In order to ensure that participants can more easily focus on working with brushing techniques and not spend too much of their time on understanding the meaning of the data attributes themselves, we decided to provide a dataset whose characteristics are easily understood in general.

Data on world trends are interesting, and the notions familiar to everyone, so we decided to carefully compile the dataset ourselves by researching freely available online data sources from, e.g., the World Bank, Internet World Stats, and the United Nations [The, Sta, Nat, Ins, Vis, Wik, UN]. The compiled data set included 84 attributes representing various country descriptors, such as summarized indexes like the Global Innovation Index (GII), the Peace Index, and the Hunger Index. In total, we included 109 countries.

In the session with the four users, we comprehensively explained the brushing techniques and numerical overlays we proposed for a more reproducible and quantitative visual analysis. We then solved one visual analysis task in which we examined a country's Global

Innovation Index Score (GII Score) to demonstrate our techniques to the participants. Figure 5.6 shows a snapshot of the interactive visual analysis (IVA) focusing on the GII, which considers a range of indicators associated with research and development, business innovation, technological readiness, and additional factors that contribute to the innovation environment of a country. In this example, to ensure that participants could focus on the linked views without being distracted by brush movement, we first utilized a percentile grid with the snap-to-grid option and then created an animated brush in Figure 5.6(a). This allowed for a smoother and more fluid brushing experience.

Following this, we discussed the displayed data correlations and the implemented statistical overviews. By providing these overviews, we were able to facilitate a more in-depth understanding of the data and highlight interesting patterns and relationships. We saw a roughly linear dependence between online creativity and internet usage per country (see the (b) scatterplot), indicating that both parameters equally influence the GII score (displayed in the (a) scatterplot). We used statistical overviews to show quantitative information for the brushed data and found that countries with a low GII score show higher variations in the percentage of internet users compared to countries with a higher GII score. We display the reference descriptive-statistics trace in black for the midrange center value in all scatterplots showing the traces. The trace of the mean center value in the (c) scatterplot confirms the linear dependence of the two data dimensions regarding the GII score. The same is confirmed using the trace of the median center value in the (d) scatterplot. However, the trace of the midrange center value, shown in the (e) scatterplot, reveals the presence of outliers. This clarifies why we observed significant variations in the relative difference plot located to the right of the (b) scatterplot. Especially in the group of countries with the highest GII score, some countries are far from the mean. The conclusion for this analysis task was that although there is a linear dependence between the GII score and online creativity, it is not important how many people in a country have internet access but how they use it.

The participants noted that the cross-hair proved to be a valuable tool, easily interpreted visually. They also appreciate the descriptive-statistics trace, as it presents more information at once than the cross-hair. They recommended including quantitative information in the table for the complete trace. After this introductory part of the demonstration, we explained to the participants how we wanted to test the reproducibility of the brushing results. The approach was that one participant should use two linked views to make an observation based on his IVA. Then, he will write a few lines of text to document his observation. Based on this text, three other users are asked to reproduce the IVA result of the first user as quickly and as accurately as possible. The text from the first user was the following: “When comparing urbanization, and the cost of living in one scatterplot, to the freedom of the press and the peace score (in a linked scatterplot), I saw that the two countries of Angola and Zambia are both among the top 20% in terms of the cost of living, as well as among the bottom 20% in terms of urbanization. I also saw that both countries, in terms of the freedom of the press and their peace scores, range within the middle of all. Can you repeat this using the percentile grid?”

## 5. DEMONSTRATION

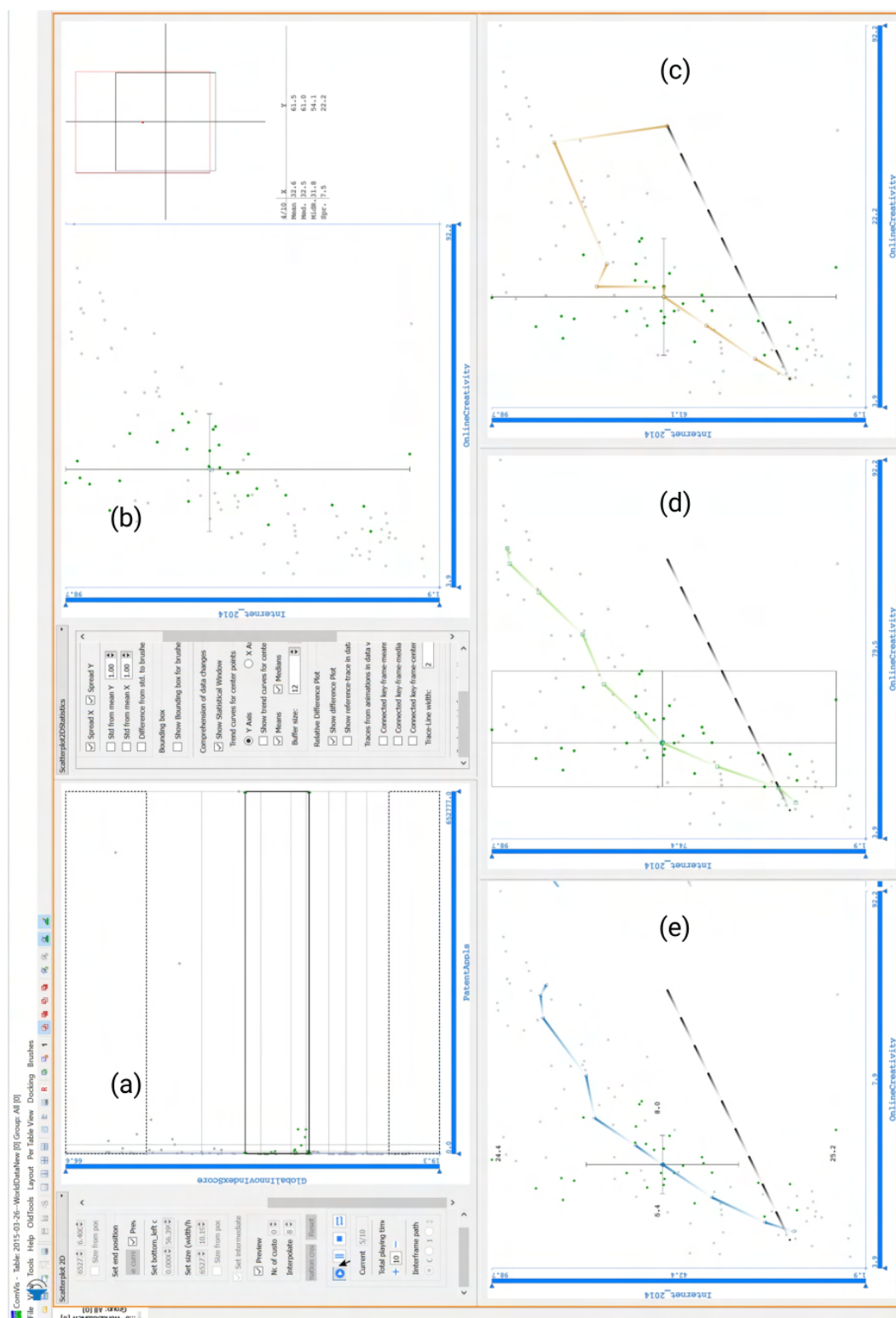


Figure 5.6: A snapshot of the introductory presentation for the participants of the second demonstration of new techniques implemented in this master thesis.

The first and the second user finished the task using the same number of steps: after showing the data in the scatterplot, both used a 20% percentile grid and a single rectangular brush to select the part of data space, which is defined by the top-left cell of the grid. Two of the three users utilized the snap-to-grid option to quickly select the data items in the top-left cell of the percentile grid, as shown in Figure 5.5. The third user also created a percentile grid in the first step, which helped him isolate the data items he needed to select, but said he felt the snap-to-grid option was not necessary here and instead created two brush combinations with a logical AND operator to select the data items of interest.

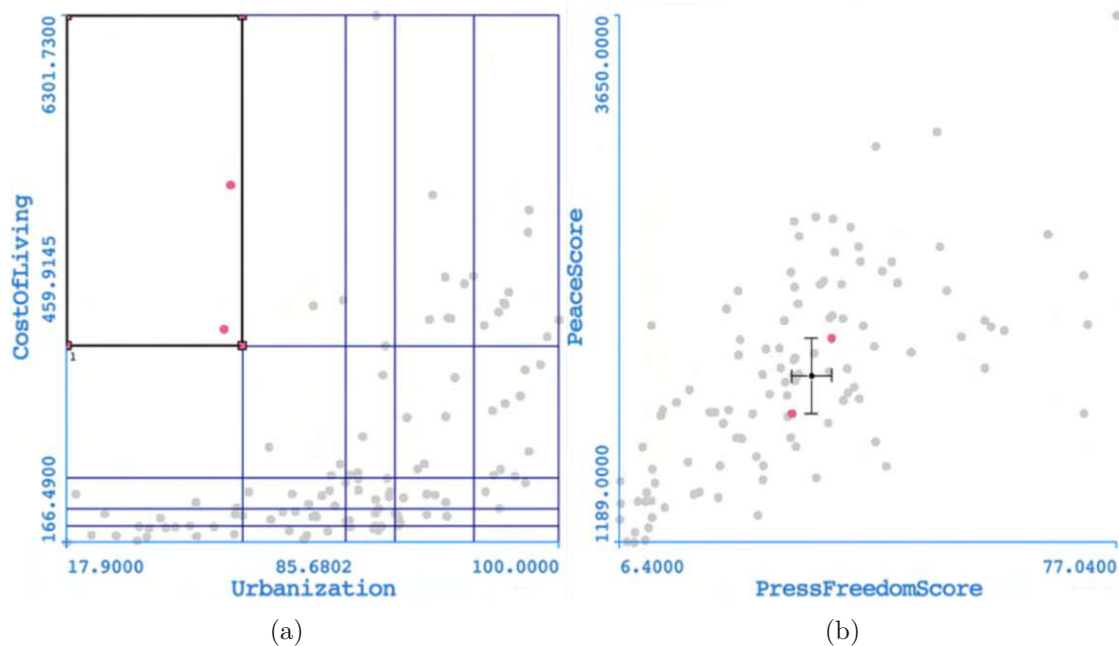


Figure 5.5: (a) A percentile grid is used as navigation tool to find data items that are in the top 20% in terms of the cost of living as well as among the bottom 20% in terms of urbanization. (b) A linked scatterplot shows the overlaid cross-hair and the median center value of the two selected data items.

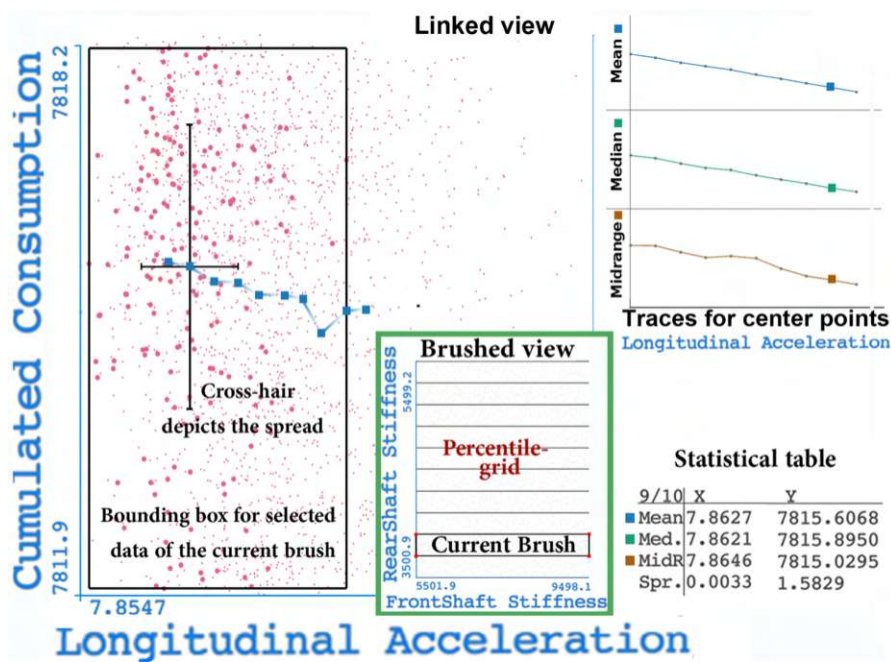
All participants indicated that the grid provided clear information and that selecting the two countries from the task was straightforward. From the visual analysis of the linked view, two of three participants concluded that the two data items are around the median center value of the entire data set. Only one user was not sure. He enabled the statistical table for quantitative inspection and stated how this was helpful to him. The easy reproducibility of the brushing results was confirmed in this case as well as in the use cases of the other three users. Another interesting example was using the circular percentile brush, where one participant asked colleagues to find the 20% of all national capitals closest to Copenhagen, Denmark's capital. Additionally, they were

asked to confirm that ten cities from that list are in the top 20% concerning the GII Score. The same user noted that descriptive statistics, together with structured brushing make the comprehension and the reporting at the end much simpler and more accessible as compared with conventional methods. He also added that the percentile brushes and grid provide a very good basis for tasks that include rank-based analysis. Encouraged by the positive feedback we received from the four participants, we aimed at publishing a research paper to inform the visualization community about our work. The next paragraph briefly discusses about the results of this undertaking.

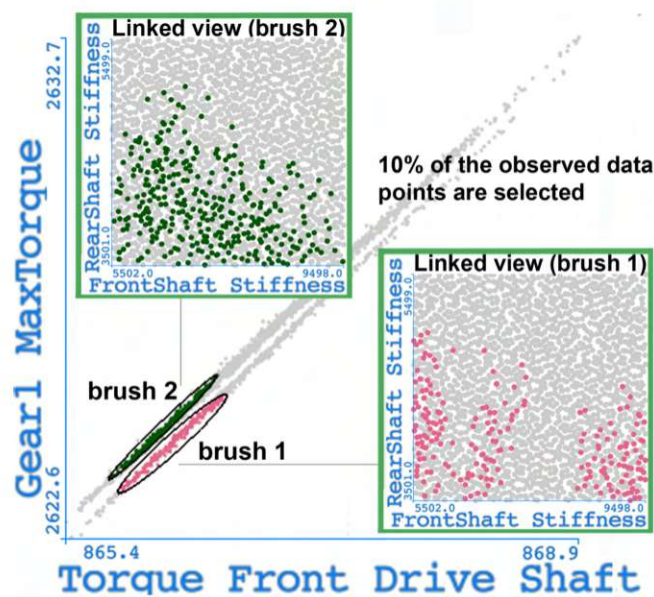
### 5.3 Related Publication

Radošs et al. [RSM<sup>+</sup>16] worked in collaboration with the domain expert to demonstrate the value of the set of constrained brushes and the novel extensions for linking&brushing, on a complex dataset from the automotive industry. The analyst noted that the brushes, which are accurately defined and supported with numerical overlays, were a valuable tool for understanding the internal relationships between fuel consumption indicators and automotive shaft design based on testing data of automobile performance. One section of the analysis is shown in Figure5.7.





(a)



(b)

Figure 5.7: (a) Brushed view (green border): The view shows a scatterplot with a percentile  $1 \times 10$  grid, and the current brush position. The brush trace is shown in the associated scatterplot on the left. Here, the distribution for the vertical-dimension was examined, which can be seen from the presented statistical table. (b) Of great help in the analysis was the Mahalanobis brush, which facilitated the examination of the structured data shown in the scatterplot.



Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar  
The approved original version of this thesis is available in print at TU Wien Bibliothek.

# Discussion and Conclusion

Exploring reproducibility and decision support through the use of advanced linking and brushing in the context of interactive visual analysis (IVA) has been a highly interesting and valuable experience. We have introduced several extensions to the IVA to improve brushing possibilities, as well as confirmed the effectiveness of quantitative extensions for decision making. In addition, we have developed the concept of a structured brushing space to support reproducibility of brushing results. In this chapter, we summarize the mayor findings and conclude the thesis. Furthermore, we present ideas for future research to expand the feature set that further improves/ensures the reproducibility and quantitative interpretation of visual analyses.

## 6.1 Research Objective IVA

We are observing a prevalent trend in which information is being conveyed through diverse visual means, encompassing both scientific works and everyday life. Examining visualizations employed outside of specialized visual analysis tools, it becomes apparent that the majority of current visualizations lack interactive features. As a result, the user's role is limited to passive observation, relying predominantly on a qualitative interpretation to gain insight based on their perceptual and cognitive abilities. If we refer to this approach as visual analysis, then the “interactive” visual analysis is a process where the user is given additional means to interact with the visualized data through a set of provided interaction techniques.

Interaction with the graphically presented data empowers users to dynamically manipulate and explore data from various perspectives, enabling them to drill down into the data, derive insights, and draw conclusions from complex and multidimensional datasets. There is a notable rise in the availability of free tools and online platforms that facilitate user interaction with the visualization system. In most cases the extent of interaction is limited to changes of data visualization techniques. On the other hand, advanced techniques,

such as interactive data selection and coordinated multiple views, have already been established as standard features in commercial visual analysis tools employed across domains such as medicine, business analytics, and various scientific fields. Unfortunately, the current approach to interactive visual analysis has certain limitations, particularly when dealing with complex and demanding data analyses.

Visual analysis can be a challenging process, requiring analysts to possess a certain level of knowledge about the data itself, as well as the available interaction and visualization techniques at their disposal. Furthermore, it necessitates intense focus and significant intellectual effort to efficiently and effectively perform interactive data exploration and analysis. Certain seemingly trivial tasks, such as reproducing brushing results and making informed decisions based on the visualized data, can often pose significant challenges or can even be impossible to accomplish. These difficulties have been observed by Kandogan et al. [KBHP14], leading to a discouragement among many new users when it comes to utilizing interactive visual analysis.

While additional assistance can be offered to users, our findings reveal that much of the previous work in this area has focused on the overall process rather than the interaction itself. The current trend in new developments is primarily aimed at further developing new visualization techniques, while comparatively less attention is paid to the improvement of interaction techniques, the consideration of the user's cognitive needs, and collaboration work. There is a great need for additional research to help users manipulate data more efficiently and accurately. Additionally, there's a need to reduce the cognitive effort required to understand the actions they perform on the data and comprehend the system's responses. These responses are primarily manifested as changes in visualizations across linked views.

In this work, we presented effective solutions for more meaningful and precise data manipulation, leading to reproducible results. By providing users with quantitative insights, we enhanced their understanding of the qualitative linked views. During the presentation of the scientific paper derived from this thesis at the EuroVis conference [RSM<sup>+</sup>16], we received numerous positive comments. Many attendees of the presentation confirmed that they had encountered issues with reproducibility in their works and expressed their desire to incorporate quantitative enhancements into their visualizations. They commended our efforts and dedication in addressing this need within the visualization community.

In this thesis, we focused on demonstrating the new techniques using only two common plot types: scatterplots and parallel coordinates. Furthermore, we specifically tailored the techniques to work with quantitative data. As a result, there is still further work to be done in implementing these techniques to other plot types and exploring their applicability to other data formats. We anticipate that interactive visual analysis and the mentioned research objectives will gain even greater importance and appeal in different fields and coming generations of data analysts as they strive to meet the increasing challenges of data analysis.

## 6.2 Human Factor Challenges in IVA

On the one hand, IVA grants users significant freedom in orchestrating the analysis process and provides powerful visualization and interaction techniques to uncover hidden insights within the analyzed data. On the other hand, IVA's efficiency and effectiveness strongly depend on individuals' perceptive and cognitive abilities to perform the analysis. The visualization community provides guidelines for tailoring visualizations and interaction techniques to accommodate the human visual system, which excels at visual sampling and interpreting qualitative data representations. However, involving users in visual analysis loop introduces numerous complexities compared to an automated computational data analysis.

Individual differences in behavior and cognitive abilities and the subjective nature of human perception increase the complexity of the design process. Therefore, default configurations are only sometimes optimal for increasingly diverse use cases or custom data that the user wants to analyze. Furthermore, users must be cautious in their selection and configuration of visualizations to ensure accurate representation of the analyzed data because improper techniques can hinder data comprehension, potentially concealing important information or leading to erroneous conclusions. During the demonstration, we observed that different users employed distinct approaches for the same task to achieve results. For example, two users made decisions based on qualitative displays, while the third user utilized an additional statistical overlay to improve his understanding of the presented data by including quantitative information. This example demonstrates that more than relying on qualitative visualization may be needed, emphasizing the importance for visualization designers to offer users resources that facilitate a quantitative interpretation of the data. Another important fact frequently overlooked is that traditional brushing techniques lack constraints, placing the direct responsibility solely on the user to maintain precise control over the brushing process. Yet, relying on the human factor to control brushing (often using a free hand) presents a challenge in IVA, as the inherent imprecision of brush usage makes it challenging to reproduce brushing results accurately. Considering the current capabilities or limitations of hardware and software, these challenges can only be fully addressed by compromising the interactive and dynamic nature of IVA. Therefore, it is crucial to reassess the current objective of having continuous reproducibility at any cost. Instead, the priority should be to support the user by providing the necessary tools to enable reproducibility whenever it is required, while minimizing any impact on the user's freedom during the analysis process.

## 6.3 Benefits of Constrained Brushing

To address the challenge of reproducing brushing results, it is essential to take into account the involvement and influence of the human factor. For our hypothesis, we assumed that the reproducibility challenge arises mostly if the brushing technique is not constrained. Unconstrained brushing allows users to utilize the brush freely according to their preferences within the defined specification of the brush in the respective view. For

instance, users can position the brush anywhere on a scatterplot and adjust the brush size from the size of a pixel to include the entire view. It is evident that utilizing an unconstrained brush, the user must carefully control the brush's anchoring, extent, and movement to achieve accurate selections of the data. The user aims to quickly finalize the data selection in the brushed view to allocate more time for analyzing changes in the linked views. Unintentional and minimal imprecision in working with the brush in the brushed view can significantly influence or change what he sees in the linked views. While brushing, users often have to repeatedly readjust the position or movement of the brush to rectify imprecisions. We demonstrated how to control the brushing interaction more precisely. Limiting brushing operations in a controlled manner, may restrict the freedom of the human analyst. Moreover, as the restrictions on the human analyst increase, the approach tends to shift more towards an automated analysis. Considering this potential limitation, we developed and implemented the concept of a structured brushing space. It is an addition to the traditional (unconstrained) IVA brushing techniques without interfering with IVA's interactive and dynamic nature. This novel approach supports the reproducibility of the analysis results by allowing the user to exert control over the main brushing operations: anchoring the brush, adjusting the extent of the brush, and controlling the movement of the brush, as well as the duration of the constraints applied. The structured brushing space is a versatile concept that can be implemented in multiple ways. We have demonstrated its feasibility through the utilization of grid elements, and constraining different brushing techniques such as the new percentile brush and a conventional rectangular brush. The advantage of our approach is that it eliminates the need for implementing a complex framework to achieve reproducibility, as typically required when capturing the entire workflow. Here, it is sufficient to make enhancements only in the visualization where the brush is used, while the other linked views remain unchanged. Therefore we named this concept "structured brushing space".

There are no limitations on the brushing techniques that can be constrained within the structured brushing space, with specific implementations tailored to the type of data and brush definition. For instance, specialized techniques may be necessary for ordered data. Furthermore, users can choose how the constraints are applied to the brush. Possibilities are to determine the desired level of restriction, ranging from unconstrained to constrained and semi-automatic, with the goal to achieve the desired brushing reproducibility. For example, users can opt to snap the brush to a standard grid for movement and utilize a percentile brush to define its extent. One of the key concerns that hinders many data analysts from using IVA is the difficulty of reproducing brushing results. If users can easily reproduce brushing results by reading comments and referring to shared screenshots, it will facilitate collaboration with colleagues and decision-making processes. We are confident that the concept of the structured brushing space with constrained brushing that enables reproducibility on-demand can significantly contribute to the widespread adoption of IVA.



## 6.4 Strengthening IVA through Quantitative Information

There are certain limitations if views provide only qualitative information, making it challenging for users to make quantitative or statistically grounded statements. In our work, we advocate for enabling quantitative interpretation of visualizations if the user needs it. We continue to support the well-known visualization mantra and testify that the qualitative nature of visual analysis is a fundamental strength as it facilitates human engagement in the analysis process. In our initial hypothesis regarding quantitative visual analysis, at the beginning of our work, we aimed to move away from statements like “we see that low values of the dimension  $x$  are correlated with high values of dimension  $y$ ” and instead provide concrete quantitative facts such as the Pearson correlation coefficient. We have chosen to define the scope of this work more clearly as it is only possible to address some of the requirements for quantitative information in the field of IVA within a single work. We decided to focus on brushing results, i.e., on selected data items, as these are often of large importance to the user, who seeks to interpret them with numerical accuracy.

One of the approaches we employed was to enhance the standard brushing technique by decorating it to display quantitative information about the brushed data items. Improving the quantitative readings of brushed data can be achieved through various enhancements, including making the brush itself quantitatively interpretable. In this respect, we introduced the percentile brush and the snap-to-grid brushing. These techniques give a clear insight into the amount of the selected data. They provide the user with a better understanding of the relative positions and rankings of the data elements, in addition to their absolute values. We also introduced enhancements to the linked views. We calculate summarized statistics for the brushed data items across various dimensions. More complex statistical methods can be used as well. These statistics are displayed in a user-friendly manner, either through a table or as an overview.

We acknowledge the presence of several unexplored opportunities for incorporating quantitative extensions into predominantly qualitative visualizations. Although this work does not include extensions specifically designed to showcase descriptive statistics for categorical data, summarized statistics can also be incorporated into visualizations that do not possess quantitative axes. Another important detail we want to highlight is the need for a thorough evaluation to examine how many quantitative additions users can interpret at once before increasing rather than reducing their mental load. The problem of visualization overplotting is closely related to the same question.

Nevertheless, our work shows that the need for more quantitative information in visualizations and extensions to support this requirement fits harmoniously with the qualitative nature of visual analysis. Both approaches, qualitative and quantitative, offer unique perspectives and insights into the data, depending on the specific analysis goals and requirements.

## 6.5 Discussion of the Evaluation

In this section, we address and comment on the considerations of domain experts regarding the conceptual and technical contributions of this thesis.

### 6.5.1 Benefits of data-driven brushes

The domain analysts responded positively to the concept of utilizing the underlying data to define more robust and potentially more meaningful brushes. They referred to the percentile brush and percentile grid, which derive statistical measures from the data. They also mentioned the Mahalanobis brush, which is quantitatively interpretable like the percentile brush, but is more advanced as it also considers the distribution direction of the underlying data structures. The idea behind these brushes involves the brush looking at the data below the brush to calculate and automatically adjust its extent to fulfill the user's specified percentage of data items to be selected. The analyst might not be accustomed to the idea of a brush changing its visual shape, which could lead to potential misinterpretation. Contrary to the initial expectation that data-driven brush type might introduce confusion, any doubts disappeared when domain experts utilized the percentile brush and the Mahalanobis brush during the demonstration. The visual changes in its extent, as the brush moves, offer supplementary insights into the distribution of data items around the mouse position. Since it is the user who controls the movements of the brush, he can effectively regulate the pace of brush movement and so the frequency of visual changes. The brushes proved advantageous for specific tasks and datasets. Statistically or data-driven brushes may have potential shortcomings. They can be overly restrictive and limit the analyst's ability to select some data items that become interesting through visual patterns. Additionally, the usefulness of these brushes might depend on the underlying data. However, this aligns perfectly with the expectations of brushing techniques in IVA, as different data may require different approaches in the analysis. Visual analysis tools typically provide standard logical operations on brushes, allowing for easy resolution of cases where a single brush is not sufficient to select complex data by allowing combinations of various brushing techniques.

All domain analysts commented that the Mahalanobis brush was extremely helpful in selecting elongated and rotated structures, as this brush automatically adjusts its orientation to capture the underlying data distribution. One analyst raised a concern that the data ranges selected by the brush are not easily interpretable. We mentioned that the ranges selected by a standard rectangular brush are easily understandable and describable if its sides are parallel to the axes in the scatterplot. It becomes more challenging to describe the selected ranges if the brush is rotated, and the same applies to the rotated ellipse used by the Mahalanobis brush. The primary purpose of the Mahalanobis brush is not range selection but rather the selection of elongated and rotated distributions that are otherwise difficult to select. The brushed items enable further inspection of the selected structures in related data dimensions in other views. Users familiar with the Mahalanobis distance will be able to understand the Mahalanobis brush easily. We

do not demonstrate a one-dimensional Mahalanobis brush, even though this is also a possible option. Having prior knowledge of the statistical methods used is beneficial for effectively utilizing advanced statistical brushes.

### 6.5.2 Off-screen Widgets and the User's Focus

Percentile brushes and the Mahalanobis brush always select a specific number of data items. The default percentage value is 10% of all data items, but users have the option to set the number according to their needs through an off-screen parameter. We refer to this parameter as off-screen because we implemented it as a widget, specifically as a field where the user enters the desired number for the percentage value. Alternatively, it could also be realized as a slider. Adjusting the brush widgets takes a person's focus away from direct data interaction, but this is referred to as indirect manipulation and it is quite common in IVA as a way for brush creation and/or manipulation.

In the case of the percentile brush, we do not see this problem being pronounced. The goal of this brush is to select a specific number of data items, and while moving the brush, it consistently selects the same percentage of all data items. The quantitative information about the brush is not lost during its movement. The use case where someone simultaneously moves the brush and changes the percentage value is indeed specific. If needed, changing the percentage value can be linked to the mouse wheel, but we have not found the need for it. With the Mahalanobis brush, we have a slightly more complex situation. In addition to the percentage parameter, we have given the user an additional auxiliary parameter through which the user can adjust the sensitivity of the brush to the local distribution of the data. This parameter can work well for a larger portion of the data. If the brush is moved to a specific location where the distribution significantly differs, the accuracy of the selection made by the brush with the existing parameter value may worsen. This requires the adjustment of this off-screen parameter at that moment. One possible alternative is to employ a clustering algorithm that automatically determines a meaningful value for the auxiliary parameter, i.e., for the sensitivity of the Mahalanobis brush. Due to the simplicity of implementation, we opted to retain our design of using off-screen widgets for parameter settings. We will briefly discuss the need for optimizing interactive techniques in the last section of the paper.

### 6.5.3 Overplotting and Occlusion

One of the goals of this work was to facilitate IVA users in obtaining quantitative information from predominantly qualitative visualizations. The first step we took was to display numeric values for the data items selected by the user in the visualization. We demonstrated several methods to accomplish this. We always had to be mindful of the occlusion problem, making sure not to hide existing data items, as well as the issue of overplotting, ensuring that the user can always view all the essential information. Although we promote the inclusion of quantitative results in the visual analysis, overlays for descriptive statistics are not enabled by default. We have three reasons for this

decision. Firstly, qualitative information predominantly provides sufficient value for initial interactive data exploration and analysis. Secondly, as mentioned earlier, descriptive statistics can obscure essential data characteristics depicted in visualizations. Thirdly, reasonable defaults for all users are hardly possible, as the selection of vital statistics depends on the specific dataset and use case. To prevent the mentioned problems, we have dedicated effort to partially mitigate them. For example, we implemented automatic positioning of numeric values for brushed data in the view, ensuring that overlapping is avoided if multiple values are shown. We also provided the option to adjust the opacity of grid lines and traces in the scatterplot or reduce the number of markers on the curve in the trace view. The initial implementation of the trace buffer in the trace view aims to address the issue of overplotting and illustrates the concept of the trace view. We recommend considering the implementation of a more advanced history overview. The advanced overview could include an independent adjustment of the distance between markers on the trace and panning and zooming capabilities, all of which help alleviate overplotting issues. Additionally, we have provided various options for configuring the display of overlaid descriptive statistics, allowing users to show and hide overlays as needed. These options include displaying descriptive statistics for all data dimensions, selected data dimensions, or displaying them on demand only. Being able to toggle the visibility of descriptive overlays quickly is important, especially when they are no longer needed or when they might obstruct a clearer view of the data visualizations.

### 6.5.4 Advancing IVA: Embracing Grids, Traces, and Animation as Standard Extensions

Currently, extensions like grids, traces, and animation are not widely available in IVA. Our work and feedback from the demonstration show their potential for a variety of new applications. Each of these extensions can be implemented independently or combined, offering new possibilities. We illustrated how these extensions can enhance brush interactions and enable reproducible and quantitative analysis, expanding IVA capabilities. Given the benefits that grids provide to users engaged in visual analysis, we recommend prioritizing the implementation of grids among these extensions. Grids are relatively simple to implement and provide numerous benefits, such as aiding analysts in data interpretation and establishing context. Moreover, we demonstrated the usage of grids in scatterplots and parallel coordinates as an intuitive tool for structuring the brushing space. Along with offering predefined default values for the grid size, such as dividing the data space into four quartiles, we provide users with the ability to set the grid based on specific tasks, either rank-based or value-based. Additionally, users have the flexibility to define non-uniform grids, further enhancing their options. We also explore the potential of automatic methods to divide the grid based on the data distribution. One of the future tasks should involve exploring methods for working with grids specifically designed for categorical data within the framework of the structured brushing space, as this aspect was not addressed in our current study.

## 6.6 Closing Statement

This thesis addresses two critical challenges in interactive visual analysis: the need for reproducibility of brushing results and the lack of quantitative information. Practical solutions to these challenges are offered by introducing new techniques such as constrained, animated, and percentile brushing. Additionally, the well-established concept of linking and brushing, which has primarily been qualitative until now, is further improved by incorporating quantitative extensions.

We develop the idea of a structured brushing space, which successfully balances the freedom of brushing with the reproducibility of results, enabling users to select the same data subset without having to record the entire workflow. We evaluate the appropriateness of grids for structuring the brushing space, and propose a snap-to-grid option for constraining brushes. Users can initiate brushing at the same position repeatedly, confine the brush to a reproducible shape, and move it along the same trajectory.

With our innovative brushing techniques such as the snap-to-grid option, animated brushing, percentile brushes, and the Mahalanobis brush, the user has the option to brush either an equal interval or an equal quantity. The Mahalanobis brush considers the local data distribution under the brush and selects a predefined number of data items. Unlike the circular percentile and standard rectangular brush, the Mahalanobis brush is particularly beneficial in regions with elongated data structures because it avoids selecting outliers from the underlying data distribution.

Utilizing the techniques, enables users to gain a deeper understanding of their interactions with the data items in the brushed view, enabling them to concentrate more efficiently on exploring linked visualizations. Animation is an example of structuring the brushing space, ensuring that selections remain simple and straightforward in the brush view while the user is free to focus on interpreting the linked view(s). Additionally, the relative difference plot complements the animation by enhancing the understanding of data changes in the linked view(s). The integration of descriptive statistics further enriches the quantitative aspect of the visual analysis.

In conclusion, we have demonstrated how new techniques can be easily integrated into existing visual analytics systems to enhance the reproducibility of brushing results and assist users in quantitatively interpreting visual analysis. These techniques help users gain better insights into data characteristics and make data-driven conclusions more effective. We have discussed the structured brushing space and introduced several brushing techniques and statistics overlays. Combined together, these techniques constitute a first major set of techniques for IVA which support reproducibility and provide quantitative insight. Although the effectiveness of our proposed techniques may vary depending on the area of application, they pave the way for further advances in linking&brushing techniques in the field of visual analysis. By enabling such advancements, visual analysis becomes even more powerful and exciting, offering a wide range of applications across diverse domains and various use cases.

### 6.7 Future Work

In this work, we have shown how to (re)design existing and implement new brushing techniques by considering the model of the structured brushing space. The structured brushing space facilitates the creation of new brushes that replicate the original brushes. This feature can promote collaboration among analysts. Thorough user studies are needed to quantify the benefits of these techniques on collaboration time. The same applies to examine how quantitative summaries and statistical plots, such as relative difference plots, can enhance comprehension of the analyzed data.

As visual analysis continues to evolve, we encourage further exploration and experimentation to expand the applicability and effectiveness of our techniques. For example, the concept of the structured brushing space can be further investigated by categorizing all common types of brushes and offering implementation suggestions based on their characteristics.

We consider it highly important to explore how visual analysis can benefit from recent advancements in AI and deep learning techniques. In this regard, one aspect that undoubtedly contributes to the improvement is the performance tuning of existing brushing techniques. One successful implementation is the work of Chan and Hauser [FH18], where they utilized the power and speed of CNN networks to implement a variant of Mahalanobis brushing. In their approach, users can mark the data items that the brush should select with a single stroke. The CNN network adjusts the brush parameters accordingly, unlike the original version of Mahalanobis brushing, where users had to adapt the brush parameters through trial and error attempts manually.

By embracing new technologies, interdisciplinary collaborations, and leveraging user feedback, we can advance the field of visual analysis and unlock its full potential in data exploration and decision support.



# Bibliography

- [198] Oxford english dictionary, 2nd ed. <https://www.dictionary.com/browse/provenance/>, Last accessed on 08-08-2023.
- [Abe13] A. Abela. *Advanced Presentations by Design: Creating Communication that Drives Action*. Wiley, 2013.
- [AHR13] Wolfgang Aigner, Stephan Hoffmann, and Alexander Rind. EvalBench: A Software Library for Visualization Evaluation. *Computer Graphics Forum*, 2013.
- [Ans73] Francis John Anscombe. Graphs in statistical analysis. *American Statistician*, 27(1):17–21, 1973.
- [Bak16] Monya Baker. 1,500 scientists lift the lid on reproducibility. *Nature*, 533:452–454, 2016.
- [BB99] Benjamin.B. Bederson and Angela. Boltman. Does animation help users build mental maps of spatial information? In *Proceedings 1999 IEEE Symposium on Information Visualization (InfoVis'99)*, pages 28–35, 1999.
- [BC87] Richard A. Becker and William S. Cleveland. Brushing scatterplots. *Technometrics*, 29(2):127–142, May 1987.
- [BH19] Leilani Battle and Jeffrey Heer. Characterizing exploratory visual analysis: A literature review and evaluation of analytic provenance in tableau. *Computer Graphics Forum*, 38(3):145–159, 2019.
- [BKKW08] Kai Burger, Polina Kondratieva, Jens Kruger, and Rudiger Westermann. Importance-driven particle techniques for flow visualization. In *2008 IEEE Pacific Visualization Symposium*, pages 71–78, March 2008.
- [BMMS91] Alexander Buja, John Alen McDonald, John Michalak, and Werner Stuetzle. Interactive data visualization using focusing and linking. In *Proceeding Visualization '91*, pages 156–163, 1991.

- [BOH11] Michael Bostock, Vadim Ogievetsky, and Jeffrey Heer. D3: Data-driven documents. *IEEE Trans. Visualization & Comp. Graphics (Proc. InfoVis)*, 2011.
- [bok] bokeh. <https://github.com/bokeh/>, Last accessed on 08-08-2023.
- [BPF14] Benjamin Bach, Emmanuel Pietriga, and Jean-Daniel Fekete. Graph diaries: Animated transitions and temporal navigation for dynamic networks. *Visualization and Computer Graphics, IEEE Transactions on*, 20(5):740–754, May 2014.
- [BS17] David Blei and Padhraic Smyth. Science and data science. *Proceedings of the National Academy of Sciences*, 114:201702076, 08 2017.
- [BSS<sup>+</sup>19] Michael Behrisch, Dirk Streeb, Florian Stoffel, Daniel Seebacher, Brian Matejek, Stefan Hagen Weber, Sebastian Mittelstädt, Hanspeter Pfister, and Daniel Keim. Commercial visual analytics systems—advances in the big data analytics field. *IEEE Transactions on Visualization and Computer Graphics*, 25(10):3011–3031, Oct 2019.
- [BTK11] Enrico Bertini, Andrada Tatu, and Daniel Keim. Quality metrics in high-dimensional data visualization: An overview and systematization. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2203–2212, 2011.
- [Cai19] Alberto Cairo. *How Charts Lie: Getting Smarter about Visual Information*. WW Norton and Co, 2019.
- [Cat] Data Visualization Catalogue. <https://datavizcatalogue.com/search.html>. <https://datavizcatalogue.com/search.html/>, Last accessed on 08-08-2023.
- [CBKK<sup>+</sup>19] Daniel Cornel, Andreas Buttinger-Kreuzhuber, Artem Konev, Zsolt Horváth, Michael Wimmer, Raimund Heidrich, and Jürgen Waser. Interactive visualization of flood and heavy rain simulations. *Computer Graphics Forum*, 38, 2019.
- [CCSK19] Akhilesh Camisetty, Chaitanya Chandurkar, Maoyuan Sun, and David Koop. Enhancing web-based analytics applications through provenance. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):131–141, 2019.
- [CGL20] Zach Cutler, Kiran Gadhave, and Alexander Lex. Ttrack: A library for provenance-tracking in web-based visualizations. In *2020 IEEE Visualization Conference (VIS)*, pages 116–120, 2020.
- [Che03] Hong Chen. Compound brushing [dynamic data visualization]. In *Information Visualization, 2003. INFOVIS 2003. IEEE Symposium on*, pages 181–188, 10 2003.

- [Cle93] William.S. Cleveland. *Visualizing Data*. At&T Bell Laboratories, 1993.
- [CM85] William S. Cleveland and Robert McGill. Graphical perception and graphical methods for analyzing scientific data. *Science*, 229:828 – 833, 1985.
- [CM88] William C. Cleveland and Marylyn E. McGill. *Dynamic Graphics for Statistics*. CRC Press, Inc., Boca Raton, FL, USA, 1st edition, 1988.
- [CMK<sup>+</sup>12] David Coyle, James Moore, Per Ola Kristensson, Paul Fletcher, and Alan Blackwell. I did that! measuring users' experience of agency in their own actions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, page 2025–2034, New York, NY, USA, 2012. Association for Computing Machinery.
- [CMS99] Stuart K. Card, Jock D. Mackinlay, and Ben Shneiderman. *Readings in Information Visualization: Using Vision to Think*. Morgan Kaufmann Publishers Inc., 1999.
- [CRM91] Stuart K. Card, George G. Robertson, and Jock D. Mackinlay. The information visualizer, an information workspace. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '91, pages 181–186, New York, NY, USA, 1991. ACM.
- [DD09] Michel Marie Deza and Elena Deza. *Encyclopedia of Distances*. Springer Berlin Heidelberg, 2009.
- [DF08] Susan Davidson and Juliana Freire. Provenance and scientific workflows: Challenges and opportunities. pages 1345–1350, 01 2008.
- [DGH03] Helmut Doleisch, Martin Gasser, and Helwig Hauser. Interactive feature specification for focus+context visualization of complex simulation data. In *Proc. of the 5th Joint IEEE TCVG - EUROGRAPHICS Symposium on Visualization (VisSym 2003)*, pages 239–248, 2003.
- [DH01] Helmut Doleisch and Helwig Hauser. Smooth brushing for focus+context visualization of simulation data in 3d. In *Journal of WSCG*, pages 147–154, 2001.
- [dV] Leonardo da Vinci. Studies of water passing obstacles. <https://www.leonardodavinci.net/>, Last accessed on 08-08-2023.
- [EDF08] Niklas Elmqvist, Pierre Dragicevic, and Jean-Daniel Fekete. Rolling the dice: Multidimensional visual exploration using scatterplot matrix navigation. *Visualization and Computer Graphics, IEEE Transactions on*, 14(6):1539–1148, Nov 2008.
- [Ekl] beeswarm: The bee swarm plot, an alternative to stripchart.

- [Eur] EuroVis. Eurovis workshop on reproducibility, verification, and validation in visualization. <https://diglib.eg.org/handle/10.2312/980/>.
- [FF20] Jean-Daniel Fekete and Juliana Freire. Exploring reproducibility in visualization. *IEEE Computer Graphics and Applications*, 40(5):108–119, 2020.
- [FFR16] Juliana Freire, Norbert Fuhr, and Andreas Rauber. Reproducibility of Data-Oriented Experiments in e-Science (Dagstuhl Seminar 16041). *Dagstuhl Reports*, 6(1):108–159, 2016.
- [FFT75] Mary Anne Fisherkeller, Jerome H. Friedman, and John W. Tukey. Prim-9: An interactive multi-dimensional data display and analysis system. In *ACM Pacific'75*, pages 140–145, 1975.
- [FH18] Chaoran Fan and Helwig Hauser. Fast and Accurate CNN-based Brushing in Scatterplots. *Computer Graphics Forum*, 2018.
- [Fis10] Danyel Fisher. *Animation for Visualization: Opportunities and Drawbacks*. O'Reilly Media, beautiful visualization edition, April 2010. Complete book available at <http://oreilly.com/catalog/0636920000617/>.
- [FKSS08] Juliana Freire, David Koop, Emanuele Santos, and Cláudio T. Silva. Provenance for computational tasks: A survey. *Computing in Science Engineering*, 10(3):11–21, 2008.
- [Fou20] Gapminder Foundation. *The UN Goals disqualify traditional ideas of progress*. Gapminder Foundation, 2020.
- [Fur86] George. W. Furnas. Generalized Fisheye Views. In *Proceedings of the ACM CHI '86 Conference on Human Factors in Computing Systems*, pages 16–23. Association for Computer Machinery, 1986.
- [FW21] Michael Friendly and Howard Wainer. *A History of Data Visualization and Graphic Communication*. Harvard University Press, 2021.
- [FWR99] Ying-Huey Fua, Matthew O. Ward, and Elke A. Rundensteiner. Hierarchical parallel coordinates for exploration of large datasets. In *VIS '99: Proceedings of the conference on Visualization '99*, pages 43–50, Los Alamitos, CA, USA, 1999. IEEE Computer Society Press.
- [FWR00] Ying-Huey Fua, Matthew O. Ward, and Elke A. Rundensteiner. Structure-based brushes: A mechanism for navigating hierarchically organized data and information spaces. *IEEE Transactions on Visualization and Computer Graphics*, 6(2):150–159, April 2000.

- [GGC<sup>+</sup>21] Kiran Gadhav, Jochen Görtler, Zach Cutler, Carolina Nobre, Oliver Deussen, Miriah Meyer, Jeff Phillips, and Alexander Lex. Predicting intent behind selections in scatterplot visualizations. *Information Visualization*, 20:147387162110386, 08 2021.
- [GL12] Steven Gomez and David Laidlaw. Modeling task performance for a crowd of users from interaction histories. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, page 2465–2468, New York, NY, USA, 2012. Association for Computing Machinery.
- [GSF<sup>+</sup>19] Florian Ganglberger, Nicolas Swoboda, Lisa Frauenstein, Joanna Kaczanowska, Wulf Haubensak, and Katja Bühler. Braintrawler: A visual analytics framework for iterative exploration of heterogeneous big brain data. *Computers & Graphics*, 06 2019.
- [GW09] David Gotz and Zhen Wen. Behavior-driven visualization recommendation. In *Proceedings of the 14th International Conference on Intelligent User Interfaces*, IUI '09, page 315–324, New York, NY, USA, 2009. Association for Computing Machinery.
- [GZL<sup>+</sup>20] Tong Ge, Yue Zhao, Bongshin Lee, Donghao Ren, Baoquan Chen, and Yunhai Wang. Canis: A high-level language for data-driven chart animations. *Computer Graphics Forum*, 39(3):607–617, 2020.
- [Hau05] Helwig Hauser. *Generalizing Focus+Context Visualization, in Scientific Visualization: The Visual Extraction of Knowledge from Data*, pages 305–327. Springer, 2005.
- [HBC<sup>+</sup>91] John Haslett, Ronan Bradley, Peter Craig, Antony Unwin, and Graham Wills. Dynamic graphics for exploring spatial data with application to locating global and local anomalies. *The American Statistician*, 45(3):234–242, 1991.
- [Hea99] Marti A. Hearst. *User Interfaces and Visualization*. Addison Wesley Longman, 1999.
- [Hin07] Jerry L. Hintze. *NCSS Statistical System*, chapter Scatter Plots with Error Bars from Summary Data. NCSS, 2007.
- [HLD02] Helwig Hauser, Florian Ledermann, and Helmut Doleisch. Angular brushing of extended parallel coordinates. In *INFOVIS '02: Proc. of the IEEE Symposium on Information Visualization (InfoVis'02)*, page 127, Washington, DC, USA, 2002. IEEE Computer Society.
- [HM] Robert B. Haber and David A. McNabb. *Visualization Idioms: A Conceptual Model for Scientific Visualization Systems*. Visualization in Scientific Computing.

- [HMSA08] Jeffrey Heer, Jock Mackinlay, Chris Stolte, and Maneesh Agrawala. Graphical histories for visualization: Supporting analysis, communication, and evaluation. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1189–1196, 2008.
- [HR07] Jeffrey Heer and George Robertson. Animated transitions in statistical data graphics. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1240–1247, November 2007.
- [HTA<sup>+</sup>15] Dandan Huang, Melanie Tory, Bon Adriel Aseniero, Lyn Bartram, Scott Bateman, Sheelagh Carpendale, Anthony Tang, and Robert Woodbury. Personal visualization and personal visual analytics. *Visualization and Computer Graphics, IEEE Transactions on*, 21:420–433, 03 2015.
- [HVF13] Samuel Huron, Romain Vuillemot, and Jean-Daniel Fekete. Visual sedimentation. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2446–2455, 2013.
- [ID90] Alfred Inselberg and Bernard Dimsdale. Parallel coordinates: a tool for visualizing multi-dimensional geometry. In *Proc. of the 1st conf. on Visualization '90*, pages 361–378, 1990.
- [Ins] International Food Policy Research Institute. <https://www.ifpri.org/>, Last accessed on 08-08-2023.
- [Ins85] Alfred Inselberg. The plane with parallel coordinates. *The Visual Computer*, 1:69–91, 1985.
- [Ins09] Alfred Inselberg. Parallel coordinates: Intelligent multidimensional visualization. 2009.
- [JF16] Jimmy Johansson and Camilla Forsell. Evaluation of parallel coordinates: Overview, categorization and guidelines for future research. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):579–588, 2016.
- [KA12] Nicholas Kong and Maneesh Agrawala. Graphical overlays: Using layered elements to aid chart reading. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2631–2638, 2012.
- [KBHP14] Eser Kandogan, Aruna Balakrishnan, Eben Haber, and Jeffrey Pierce. From data to insight: Work practices of analysts in the enterprise. *Computer Graphics and Applications, IEEE*, 34(5), 2014.
- [KCD<sup>+</sup>09] Nazanin Kadivar, Victor Chen, Dustin Dunsmuir, Eric Lee, Cheryl Qian, John Dill, Christopher Shaw, and Robert Woodbury. Capturing and supporting the analysis process. In *2009 IEEE Symposium on Visual Analytics Science and Technology*, pages 131–138, 2009.



- [KCH19] Younghoon Kim, Michael Correll, and Jeffrey Heer. Designing animated transitions to convey aggregate operations. *Computer Graphics Forum*, 38:541–551, 06 2019.
- [KFH10] Johannes Kehrler, Peter Filzmoser, and Helwig Hauser. Brushing moments in interactive visual analysis. In *Proceedings of the 12th Eurographics / IEEE - VGTC Conference on Visualization*, EuroVis’10, pages 813–822, Aire-la-Ville, Switzerland, Switzerland, 2010. Eurographics Association.
- [KH21] Younghoon Kim and Jeffrey Heer. Gemini: A grammar and recommender system for animated transitions in statistical graphics. *IEEE Transactions on Visualization and Computer Graphics*, 27:485–494, 2021.
- [KKEM10] Daniel Keim, Jörn Kohlhammer, Geoffrey Ellis, and Florian Mansmann. *Mastering the Information Age. Solving Problems with Visual Analytics*. Eurographics Association, Goslar, 2010.
- [KLM<sup>+</sup>12] Zoltán Konyha, Alan Lež, Krešimir Matković, Mario Jelović, and Helwig Hauser. Interactive visual analysis of families of curves using data aggregation and derivation. In *Proceedings of the 12th International Conference on Knowledge Management and Knowledge Technologies*, 2012.
- [KM16] Rob Kitchin and Gavin McArdle. What makes big data, big data? exploring the ontological characteristics of 26 datasets. *Big Data & Society*, 3(1):2053951716631130, 2016.
- [KMG<sup>+</sup>06] Zoltán Konyha, Krešimir Matković, Denis Gračanin, Mario Jelović, and Helwig Hauser. Interactive visual analysis of families of function graphs. *IEEE Transactions on Visualization and Computer Graphics*, 12(6):1373–1385, 2006.
- [KMS<sup>+</sup>08] Daniel A. Keim, Florian Mansmann, Jörn Schneidewind, Jim Thomas, and Hartmut Ziegler. *Visual Analytics: Scope and Challenges*, pages 76–90. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.
- [KMSZ06] Daniel A. Keim, Florian Mansmann, Jörn Schneidewind, and Hartmut Ziegler. Challenges in visual data analysis. In *In Proceedings of the Tenth International Conference on Information Visualization*, pages 9–16, 2006.
- [Knu84] Donald E. Knuth. Literate programming. *Comput. J.*, 27(2):97–111, may 1984.
- [Kos] Rober Kosara. <https://eagereyes.org/>, Last accessed on 08-08-2023.
- [KPV<sup>+</sup>18] Phillip Koytek, Charles Perin, Jo Vermeulen, Elisabeth André, and Sheelagh Carpendale. Mybrush: Brushing and linking with personal agency. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):605–615, Jan 2018.

- [KS22] Elif Korkut and Elif Surer. Visualization in virtual reality: a systematic review, 03 2022.
- [LAB<sup>+</sup>06] Bertram Ludäscher, Ilkay Altintas, Chad Berkley, Dan Higgins, Efrat Jaeger, Matthew Jones, Edward A. Lee, Jing Tao, and Yang Zhao. Scientific workflow management and the kepler system: Research articles. *Concurr. Comput.: Pract. Exper.*, 18(10):1039–1065, aug 2006.
- [LFW<sup>+</sup>20] Min Lu, Noa Fish, Shuaiqi Wang, Joel Lanir, Daniel Cohen-Or, and Hui Huang. Enhancing static charts with data-driven animations. *IEEE Transactions on Visualization and Computer Graphics*, pages 1–1, 2020.
- [LIG] LIGO. Analyzing gravitational wave data from the ligo open science center. <https://rmarkdown.rstudio.com/>.
- [LKD19] Ricardo Langner, Ulrike Kister, and Raimund Dachselt. Multiple coordinated views at large displays for multiple users: Empirical findings on user behavior, movements, and distances. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):608–618, 2019.
- [LMS<sup>+</sup>18] Deqing Li, Honghui Mei, Yi Shen, Shuang Su, Wenli Zhang, Junting Wang, Ming Zu, and Wei Chen. Echarts: A declarative framework for rapid construction of web-based visualization. *Visual Informatics*, 2, 05 2018.
- [LMvW10] Jing Li, Jean-Bernard Martens, and Jarke J. van Wijk. A model of symbol size discrimination in scatterplots. In *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems*, CHI '10, page 2553–2562, New York, NY, USA, 2010. Association for Computing Machinery.
- [LPVM15] Ji Liu, Esther Pacitti, Patrick Valduriez, and Marta Mattoso. A survey of data-intensive scientific workflow management. *Journal of Grid Computing*, 13, 03 2015.
- [M18] Thomas Mühlbacher. *Human-Oriented Statistical Modeling: Making Algorithms Accessible through Interactive Visualization*. PhD thesis, "Institute of Computer Graphics and Algorithms, Vienna University of Technology ", "Favoritenstrasse 9-11/E193-02, A-1040 Vienna, Austria", aug "2018".
- [Mah36] Prasanta Chandra Mahalanobis. On the generalized distance in statistics. *Proceedings of the National Institute of Sciences (Calcutta)*, 2:49–55, 1936.
- [MDB87] Bruce H. McCormick, Thomas A. DeFanti, and Maxine D. Brown. Visualization in scientific computing. *Computer Graphics*, vol 21(6), November 1987.

- [MFGH08] Krešimir Matković, Wolfgang Freiler, Denis Gračanin, and Helwig Hauser. Comvis: a coordinated multiple views system for prototyping new visualization technology. In *Proceedings of the 12th International Conference Information Visualisation*, pages
- [MGH18] Krešimir Matković, Denis Gračanin, and Helwig Hauser. Visual analytics for simulation ensembles. In *2018 Winter Simulation Conference (WSC)*, pages 321–335, 2018.
- [MGWM21] Jimmy Moore, Pascal Goffin, Jason Wiese, and Miriah Meyer. Exploring the personal informatics analysis gap: "there's a lot of bacon". 08 2021.
- [MHSW19] Eva Mayr, Nicole Hynek, Saminu Salisu, and Florian Windhager. Trust in Information Visualization. In Robert Kosara, Kai Lawonn, Lars Linsen, and Noeska Smit, editors, *EuroVis Workshop on Trustworthy Visualization (TrustVis)*. The Eurographics Association, 2019.
- [Min69] Charles Joseph Minard. Napoleon's march to moscow, 1869. Courtesy of Tufte, Edward R. Beautiful Evidence. Cheshire, Conn: Graphics Press, 2006.
- [MMWC18] Honghui Mei, Yuxin Ma, Yating Wei, and Wei Chen. The design space of construction tools for information visualization: A survey. *Journal of Visual Languages & Computing*, 44:120–132, 2018.
- [MRRS18] Golam Mostaeen, Banani Roy, Chanchal K. Roy, and Kevin A. Schneider. Fine-grained attribute level locking scheme for collaborative scientific workflow development. In *2018 IEEE International Conference on Services Computing (SCC)*, pages 273–277, 2018.
- [Mun10] Tamara Munzner. A nested model for visualization design and validation. *Visualization and Computer Graphics, IEEE Transactions on*, 15:921 – 928, 01 2010.
- [Mun14] Tamara Munzner. *Visualization Analysis and Design*. AK Peters Visualization Series. CRC Press, 2014.
- [MW95] Alen R. Martin and Matthew O. Ward. High dimensional brushing for interactive exploration of multivariate data. In *Visualization, 1995. Visualization '95. Proceedings., IEEE Conference on*, pages 271–, 1995.
- [Nat] NationMaster. [https://https://www.nationmaster.com/](https://www.nationmaster.com/), Last accessed on 08-08-2023.
- [Nig] Florence Nightingale. Diagram of the causes of mortality in the army in the east. Technical report, National Army Museum. Courtesy of National Army Museum, United Kingdom.

- [NOA14] NOAA. National Climatic Data Center, 2014.
- [Not] Jupyter Notebooks. <https://jupyter.org/>, Last accessed on 08-08-2023.
- [NS97] Chris North and Ben Shneiderman. A taxonomy of multiple window coordination. Technical report, School of Computing, University of Maryland: College Park, MD, USA, 1997.
- [NXW<sup>+</sup>16] Phong H. Nguyen, Kai Xu, Ashley Wheat, B.L. William Wong, Simon Attfeld, and Bob Fields. Sensepath: Understanding the sensemaking process through analytic provenance. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):41–50, 2016.
- [Nø98] Tor Nørretranders. *The user illusion: Cutting consciousness down to size*. Viking, 1998.
- [OAF<sup>+</sup>04] Tom Oinn, Matthew Addis, Justin Ferris, Darren Marvin, Martin Senger, Mark Greenwood, Tim Carver, Kevin Glover, Matthew Pocock, Anil Wipat, and Peter Li. Taverna: A tool for the composition and enactment of bioinformatics workflows. *Bioinformatics (Oxford, England)*, 20:3045–54, 12 2004.
- [Obs] Observable. <https://observablehq.com/>, Last accessed on 08-08-2023.
- [OJ15] Mershack Okoe and Radu Jianu. Graphunit: Evaluating interactive graph visualizations using crowdsourcing. *Computer Graphics Forum*, 34, 06 2015.
- [oSEM19] National Academies of Sciences Engineering and Medicine. *Reproducibility and Replicability in Science*. The National Academies Press, Washington, DC, 2019.
- [pb] power bi. <https://powerbi.microsoft.com/en-us/what-is-power-bi/>, Last accessed on 08-08-2023.
- [PCW89] Antony Unwin Peter Craig, John Haslett and Graham Wills. Moving statistics: An extension of brushing for spatial data. In *Proceedings of the 21st Symposium on the Inreface*, page 170–174. Science and Statistics, 1989.
- [plo] plotly. <https://github.com/plotly/>, Last accessed on 08-08-2023.
- [PNH<sup>+</sup>20] Q. Pham, D. C. Nguyen, T. Huynh-The, W. Hwang, and P. N. Pathirana. Artificial intelligence (ai) and big data for coronavirus (covid-19) pandemic: A survey on the state-of-the-arts. *IEEE Access*, 8:130820–130839, 2020.
- [PWR04] Wei Peng, M.O. Ward, and E.A. Rundensteiner. Clutter reduction in multi-dimensional data visualization using dimension reordering. In *IEEE Symposium on Information Visualization*, pages 89–96, 2004.

- [RAM<sup>+</sup>11] Alexander Rind, Wolfgang Aigner, Silvia Miksch, Sylvia Wiltner, Margit Pohl, Felix Drexler, Barbara Neubauer, and Nikolaus Suchy. Visually exploring multivariate trends in patient cohorts using animated scatter plots. In Michelle M. Robertson, editor, *Ergonomics and Health Aspects of Work with Computers*, pages 139–148, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.
- [RBR22] Jonathan Roberts, Peter Butcher, and Panagiotis Ritsos. One view is not enough: Review of and encouragement for multiple and alternative representations in 3d and immersive visualisation. *Computers*, 11:20, 02 2022.
- [RCM<sup>+</sup>20] Guido Reina, Hank Childs, Krešimir Matković, Katja Bühler, Manuela Waldner, David Pugmire, Barbora Kozlíková, Timo Ropinski, Patric Ljung, Takayuki Itoh, Eduard Gröller, and Michael Krone. The moving target of visualization software for an increasingly complex world. *Computers and Graphics*, 87:12–29, 2020.
- [RESC16] Eric D. Ragan, Alex Endert, Jibonananda Sanyal, and Jian Chen. Characterizing provenance in visualization and data analysis: An organizational framework of provenance types and purposes. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):31–40, 2016.
- [RFF<sup>+</sup>08] George Robertson, Roland Fernandez, Danyel Fisher, Bongshin Lee, and John Stasko. Effectiveness of animation in trend visualization. *Visualization and Computer Graphics, IEEE Transactions on*, 14(6):1325–1332, Nov 2008.
- [RGR17] David Reinsel, John Gantz, and John Rydning. *Data Age 2025: The Evolution of Data to Life-Critical Don't Focus on Big Data; Focus on the Data That's Big (white paper)*. 2017.
- [RLA<sup>+</sup>19] Richard C. Roberts, Robert S. Laramee, Smith Gary A., Paul Brookes, and Tony D'Cruze. Smart brushing for parallel coordinates. *IEEE Transactions on Visualization and Computer Graphics*, 25(3):1575–1590, 2019.
- [RLB19] Donghao Ren, Bongshin Lee, and Matthew Brehmer. Charticulator: Interactive construction of bespoke chart layouts. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):789–799, jan 2019.
- [RMa] RMarkdown. <https://rmarkdown.rstudio.com/>, Last accessed on 08-08-2023.
- [Rob07] Jonathan C. Roberts. State of the art: Coordinated & multiple views in exploratory visualization. In Gennady Andrienko, Jonathan C. Roberts, and Chris Weaver, editors, *Proc. of the 5th International Conference on Coordinated & Multiple Views in Exploratory Visualization*. IEEE CS Press, 2007.

- [RSM<sup>+</sup>16] S. Radoš, R. Splechtna, K. Matković, M. Đuras, E. Gröller, and H. Hauser. Towards quantitative visual analytics with structured brushing and linked statistics. *Computer Graphics Forum*, 35(3):251–260, 2016.
- [SBG<sup>+</sup>18] Rainer Splechtna, Michael Beham, Denis Gračanin, M. Ganuza, Katja Bühler, Igor Pandžić, and Krešimir Matković. Cross-table linking and brushing: interactive visual analysis of multiple tabular data sets. *The Visual Computer*, 34, 06 2018.
- [SCB<sup>+</sup>19] Alper Sarikaya, Michael Correll, Lyn Bartram, Melanie Tory, and Danyel Fisher. What do we talk about when we talk about dashboards? *IEEE Transactions on Visualization and Computer Graphics*, 25(1):682–692, 2019.
- [SDE<sup>+</sup>16] Rainer Splechtna, Alexandra Diehl, Mai Elshehaly, Claudio Delrieux, Denis Gračanin, and Krešimir Matković. Bus lines explorer: Interactive exploration of public transportation data. In *Proceedings of the 9th International Symposium on Visual Information Communication and Interaction, VINCI '16*, page 30–34, New York, NY, USA, 2016. Association for Computing Machinery.
- [SFSA10] Cláudio Silva, Juliana Freire, Emanuel Santos, and Erik Anderson. Provenance-enabled data exploration and visualization with vistrails. In *Graphics, Patterns and Images Tutorials (SIBGRAPI-T), 2010 23rd SIBGRAPI Conference on*, pages 1–9, 8 2010.
- [SGMS21] Raphael Sahann, Ivana Gajic, Torsten Moeller, and Johanna Schmidt. Selective Angular Brushing of Parallel Coordinate Plots. In Marco Agus, Christoph Garth, and Andreas Kerren, editors, *EuroVis 2021 - Short Papers*. The Eurographics Association, 2021.
- [SH14] Arvind Satyanarayan and Jeffrey Heer. Lyra: An Interactive Visualization Design Environment. *Computer Graphics Forum (Proc. EuroVis)*, 2014.
- [Shn] Ben Shneiderman.
- [Shn92] Ben Shneiderman. Tree visualization with tree-maps: 2-d space-filling approach. *ACM Trans. Graph.*, 11(1):92–99, January 1992.
- [Shn96] Ben Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Proc. of the 1996 IEEE Symp. on Visual Languages*, page 336, 1996.
- [SI08] Maruthappan Shanmugasundaram and Pourang Irani. The effect of animated transitions in zooming interfaces. In *AVI '08*, 2008.
- [SKGM21] Disha Sardana, Sampanna Kahu, Denis Gračanin, and Krešimir Matković. *Multi-modal Data Exploration in a Mixed Reality Environment Using Coordinated Multiple Views*, pages 337–356. 07 2021.



- [SMWH17] Arvind Satyanarayan, Dominik Moritz, Kanit Wongsuphasawat, and Jeffrey Heer. Vega-lite: A grammar of interactive graphics. *IEEE Transactions on Visualization & Computer Graphics (Proc. InfoVis)*, 2017.
- [Spe87] Charles Spearman. The proof and measurement of association between two things. By C. Spearman, 1904. *The American journal of psychology*, 100(3-4):441–471, 1987.
- [SR06] Harri Siirtola and Kari-Jouko Räihä. Interacting with parallel coordinates. *Interacting with Computers*, 18(6):1278–1309, 06 2006.
- [SS16] Ramik Sadana and John T. Stasko. Designing multiple coordinated visualizations for tablets. *Comput. Graph. Forum*, 35(3):261–270, 2016.
- [SS19] Hayeong Song and Danielle Albers Szafir. Where’s my data? evaluating visualizations with missing data. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):914–924, 2019.
- [Sta] Internet World Stats. [https://https://www.internetworldstats.com/](https://www.internetworldstats.com/), Last accessed on 08-08-2023.
- [Sta14] John Stasko. Value-driven evaluation of visualizations. In *Proceedings of the Fifth Workshop on Beyond Time and Errors: Novel Evaluation Methods for Visualization*, BELIV ’14, page 46–53, New York, NY, USA, 2014. Association for Computing Machinery.
- [Tab20] Tableau. Tableau tool, 2020. <https://public.tableau.com/s/>, Last accessed on 08-08-2023.
- [TC05] James J. Thomas and Kristin A. Cook. *Illuminating the Path: The Research and Development Agenda for Visual Analytics*. National Visualization and Analytics Ctr, 2005.
- [THC19] Christoph Traxler, Gerd Hesina, and Klaus Chmelina. *Immersive tunnel monitoring by data driven navigation in 3D*, pages 3254–3261. 04 2019.
- [The] The World Bank. [https://https://data.worldbank.org/](https://data.worldbank.org/), Last accessed on 08-08-2023.
- [TLCV15] Chun-Wei Tsai, C. Lai, H. Chao, and A. Vasilakos. Big data analytics: a survey. *Journal of Big Data*, 2:1–32, 2015.
- [TM87] Paul Tukey and Vonn Marsch. Discovering features of 3-dimensional data through computer animation using ”plot 3-d”, 1987. <https://community.amstat.org/jointscsg-section/media/videos/>, Last accessed on 08-08-2023.

- [TMB02] Barbara Tversky, Julie Bauer Morrison, and Mireille Betrancourt. Animation: Can it facilitate? *Int. J. Hum.-Comput. Stud.*, 57(4):247–262, October 2002.
- [TT06] Susan Bell Trickett and J. Gregory Trafton. Toward a comprehensive model of graph comprehension: Making the case for spatial cognition. In Dave Barker-Plummer, Richard Cox, and Nik Swoboda, editors, *Diagrammatic Representation and Inference*, pages 286–300, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.
- [Tuf] Edward R. Tufte. <http://www.edwardtufte.com/tufte/>, Last accessed on 08-08-2023.
- [Tuf83] Edward R. Tufte. *The Visual Display Of Quantitative Information*. Graphics Press, 1983.
- [Tuk77] John W. Tukey. *Exploratory Data Analysis*. Addison-Wesley, Reading, MA, 1977.
- [tV] From Data to Viz. <https://www.data-to-viz.com/>, Last accessed on 08-08-2023.
- [UN] UN. <https://www.un.org/en/>, Last accessed on 08-08-2023.
- [Vis] Vision of Humanity. <https://www.visionofhumanity.org/>, Last accessed on 08-08-2023.
- [VRV] VRVis. <http://www.VRVis.at/vis/>, Last accessed on 08-08-2023.
- [VSOC21] Milena Vuckovic, Johanna Schmidt, Thomas Ortner, and Daniel Cornel. Combining 2d and 3d visualization with visual analytics in the environmental domain. *Information (Switzerland)*, 13, 12 2021.
- [War94] Matthew O. Ward. Xmdvtool: integrating multiple methods for visualizing multivariate data. In *VIS '94: Proceedings of the conference on Visualization '94*, pages 326–333, Los Alamitos, CA, USA, 1994. IEEE Computer Society Press.
- [War22] Colin Ware. *Contents*. Morgan Kaufmann, second edition edition, 2022.
- [WGK15] Matthew O. Ward, Georges Grinstein, and Daniel Keim. *Interactive Data Visualization: Foundations, Techniques, and Applications, Second Edition*. A K Peters/CRC Press, 2015.
- [WGK21] Matthew O. Ward, Georges Grinstein, and Daniel Keim. *Interactive Data Visualization Foundations, Techniques, and Applications, Second Edition*. CRC Press, 2021.

- [WH14] Gunther H. Weber and Helwig Hauser. Interactive visual exploration and analysis. In Charles D. Hansen, Min Chen, Chris R. Johnson, Arie E. Kaufman, and Hans Hagen, editors, *Scientific Visualization: Uncertainty, Multifield, Bio-Medical and Scalable Visualization*, Mathematics and Visualization, pages 161–174. Springer-Verlag, 2014. LBNL-6655E.
- [Wik] Wikipedia. <https://en.wikipedia.org/wiki/Wikipedia/>, Last accessed on 08-08-2023.
- [Wil96] G. J. Wills. Selection: 524,288 ways to say "this is interesting". In *Proceedings of the 1996 IEEE Symposium on Information Visualization (INFOVIS '96)*, INFOVIS '96, pages 54–, Washington, DC, USA, 1996.
- [Wil05] Leland Wilkinson. *The Grammar of Graphics (Statistics and Computing)*. Springer-Verlag, Berlin, Heidelberg, 2005.
- [WJXN16] Eugene Wu, Lilong Jiang, Larry Xu, and Arnab Nandi. Graphical perception in animated bar charts. 03 2016.
- [XOW<sup>+</sup>20] Kai Xu, Alvitta Ottley, Conny Walchshofer, Marc Streit, Remco Chang, and John Wenskovitch. Survey on the analysis of user interactions and visualization provenance. *Computer Graphics Forum*, 39(3):757–783, 2020.
- [XS20] Yushun Xiao and Qi Sun. A new visualization for many-objective optimization. In *2020 5th International Conference on Mechanical, Control and Computer Engineering (ICMCCE)*, pages 1998–2002, 2020.
- [ZKL14] Jia Zhang, Daniel Kuc, and Shiyong Lu. Confucius: A tool supporting collaborative scientific workflow composition. *IEEE Transactions on Services Computing*, 7(1):2–17, 2014.